Институт системного программирования
Российской академии наук

В.В. Липаев

ПРОЕКТИРОВАНИЕ И ПРОИЗВОДСТВО СЛОЖНЫХ ЗАКАЗНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

СИНТЕГ
Москва - 2011

УДК 004.41(075.8)
ББК 32.973.26-018я73
Л61

Липаев В.В.
Проектирование и производство сложных заказных программных продуктов. – М.: СИНТЕГ, 2011. – 408 с.
ISBN 978-5-89638-119-8

Монография состоит из двух частей, в которых изложены методы и процессы проектирования и производства сложных заказных программных продуктов для технических систем реального времени. Все компоненты и комплексы программ должны соответствовать требованиям заказчика, высокому качеству и минимальным рискам посредством верификации, тестирования, испытаний и сертификации, обеспечиваемыми коллективами квалифицированных специалистов. При изложении активно используются современные международные и отечественные стандарты, планирование производственных процессов с учетом ограниченных экономических ресурсов крупных проектов.

Часть 1 посвящена методам системного проектирования комплексов программ, подбору и подготовке коллектива специалистов для проектирования и производства сложных программных продуктов. Изложено проектирование требований к компонентам и комплексам программ, а также требований к характеристикам качества и допустимым рискам при проектировании процессов производства программных комплексов. Представлено оценивание и прогнозирование сложности проектирования и экономических характеристик процессов производства заказных программных продуктов.

Часть 2 содержит основы промышленного производства сложных заказных программных продуктов. Изложены организация и реализация верификации и тестирования комплексов программ, тестирования потоков управления и потоков данных программных модулей и компонентов, планирование производства и тестирования компонентов и комплексов программ. Представлено тестирование сложных динамических программных продуктов и методы сопровождения программных комплексов. Изложены методы и процессы управления конфигурацией и документированием программных комплексов, а также испытания, удостоверение качества и сертификация сложных заказных программных продуктов с учетом стандартов.

Монография предназначена для руководителей предприятий и проектов технических систем, для специалистов, ответственных за проектирование и производство сложных заказных программных продуктов реального времени высокого качества, также может использоваться в качестве учебного пособия по программной инженерии.

ОГЛАВЛЕНИЕ

Глава 1.5. Прогнозирование сложности проектирования заказных программных продуктов ……

Основные факторы, определяющие сложность проектирования заказных программных продуктов. Прогнозирование сложности проектирования процессов производства заказных программных продуктов на основе экономических характеристик. Характеристики трудоемкости и длительности проектирования процессов производства программных продуктов.

Глава 1.6. Прогнозирование экономических характеристик процессов производства заказных программных продуктов ……..

Простейшие модели прогнозирования экономических характеристик производства программных продуктов. Модель прогнозирования экономических характеристик проектирования производства программных продуктов COCOMO II. Влияние масштабных факторов при прогнозировании экономических характеристик производства программных продуктов. Влияние коллектива специалистов при прогнозировании экономических характеристик проектирования и производства программных продуктов. Влияние технологической и компьютерной среды проектирования и производства при прогнозировании экономических характеристик программных продуктов. Средства автоматизации прогнозирования экономических характеристик производства программных продуктов.

Часть 2. Производство заказных программных продуктов …

Глава 2.1. Основные производственные процессы сложных заказных комплексов программ ……...

Стандарты производственных процессов сложных комплексов программ. Производственные процессы обеспечения качества компонентов и комплексов программ. Производственные процессы верификации и тестирования сложных комплексов программ. Производственные процессы документирования сложных комплексов программ. Дефекты и ошибки в компонентах и сложных комплексах программ.

Глава 2.2. Организация верификации и тестирования компонентов и комплексов программ ……

Процессы верификации компонентов и комплексов программ. Трассирование взаимодействия требований к компонентам в комплексах программ. Организация процессов тестирования компонентов и комплексов программ. Процессы и методы тестирования программных модулей и компонентов.

Глава 2.3. Тестирование потоков управления и потоков данных заказных программных модулей и компонентов …….
Стратегии выбора тестов потоков управления для программных модулей. Сложность тестирования потоков управления программных модулей. Корректность тестирования потоков управления программных модулей. Тестирование потоков данных программных модулей. Тестирование модулей программ с учетом значений переменных и констант.

Глава 2.4. Планирование производства и тестирования заказных компонентов и комплексов программ ……
Планирование производства компонентов для комплексов программ. Графики для планирования производства программных продуктов. Стратегии систематического тестирования сложных комплексов программ. Планирование тестирования сложных заказных компонентов и комплексов программ. Программа, график разработки и выполнения тестов для сложных комплексов программ. Особенности планирования тестирования сложных заказных комплексов программ.

Глава 2.5. Тестирование при производстве сложных заказных динамических программных продуктов ……..
Подготовка динамических тестов для тестирования сложных заказных программных продуктов. Компоненты генераторов динамических тестов внешней среды в реальном времени. Обработка результатов динамического тестирования комплексов программ в реальном времени. Тестирование надежности и безопасности функционирования программных продуктов в реальном времени. Тестирование производительности и использования ресурсов компьютеров программными продуктами.

Глава 2.6. Сопровождение сложных заказных программных комплексов ……
Организация и методы сопровождения сложных программных комплексов. Этапы и процедуры при сопровождении сложных заказных программных комплексов. Ресурсы, для обеспечения сопровождения сложных заказных программных комплексов.

Глава 2.7. Управление конфигурацией и документирование сложных заказных программных комплексов ……..
Процессы управления конфигурацией программных комплексов. Этапы и процедуры при управлении конфигурацией заказных программных комплексов. Организация документирования заказных программных комплексов. Подготовка эксплуатационной документации для заказных программных продуктов.

ПРЕДИСЛОВИЕ

Уважаемые читатели, в результате подвижнического труда Владимира Васильевича Липаева появилась очередная книга, продолжающая фундаментальный цикл его работ по теории и практике создания сложных программных комплексов больших технических систем, функционирующих в квазиреальном или реальном масштабе времени. С большой глубиной в настоящей книге автором исследуются все этапы жизненного цикла сложных заказных программных продуктов, которые разрабатываются по тактико-техническим заданиям в интересах создания оборонных систем управления и обработки информации, систем управления воздушным движением, космических систем дистанционного зондирования Земли и им подобных.

Важное практическое значение имеют рассмотренные в книге вопросы выработки требований к характеристикам качества при допустимом уровне риска, прогнозирования сложности проектирования, оценки экономических показателей и организации больших коллективов разработчиков сложных заказных программных продуктов.

Создание таких продуктов промышленными методами и многочисленными коллективами специалистов определяет необходимость четкого планирования работ как по ресурсам, так и по этапам и срокам. Не менее важно в ходе их производства четко организовать процессы верификации и тестирования программных компонентов и программного комплекса в целом, включая тестирование потоков управления и потоков входных данных. Разделы книги, посвященные этим вопросам, имеют, на мой взгляд, большую практическую ценность, подкрепляемую огромным личным опытом автора.

Сопровождение сложных заказных программных продуктов – это один из самых трудоемких этапов их жизненного цикла. Автор уделяет этому пристальное внимание, показывая, что реализация этапа сопровождения не может быть успешной без тщательного документирования, организованного автоматизированным способом.

Быстрый рост сложности и повышения требований к качеству программных продуктов привели к необходимости дифференцированного обучения специалистов в области программной инженерии, разделения труда при выполнении различных видов работ в ходе соз-

дания программных продуктов, проведения детального экономического анализа, а также многоуровневого планирования деятельности больших производственных коллективов. Для организации эффективной структуры таких коллективов их руководители должны учитывать и согласовывать конкретные цели участвующих в крупном проекте специалистов, психологическую совместимость, способность к согласованной работе, опыт взаимодействия, а также другие объективные и субъективные свойства участников проекта. Этим вопросам автор уделяет внимание, хотя они, по моему мнению, нуждаются в проведении самостоятельных исследований.

Я надеюсь, что отражу общее мнение генеральных конструкторов больших технических систем, функционирующих в реальном времени, выражая глубокую благодарность автору за многочисленные исследования по различным аспектам теории и практики создания сложных заказных программных продуктов. В этом можно убедиться при ознакомлении в конце книги с обширным перечнем трудов, опубликованных уважаемым Владимиром Васильевичем Липаевым на протяжении нескольких десятилетий.

Генеральный директор - генеральный конструктор
ФГУП «ЦНИИ «Комета», доктор технических наук,
профессор **Мисник В.П.**

ВВЕДЕНИЕ

Для организации проектирования и производства ***сложных заказных программных продуктов*** целесообразно ***использовать в качестве ориентиров*** современные индустриальные методы и процессы в области промышленного создания сложных технических систем и продуктов. Поэтому в предисловии кратко изложены общие основы организации производства технических систем. Их основные положения представлены в предисловии для применения при проектировании и производстве сложных систем и программных продуктов. В монографии не всегда явно подразумевается их применение в течение жизненного цикла ***заказных программных продуктов*** реального времени, что ограничивает и конкретизирует область рассматриваемых комплексов программ.

Массовое создание сложных и дорогих технических систем и продуктов промышленными методами и большими коллективами специалистов определяет необходимость четкой организации проектирования и производства, таких работ по этапам и срокам реализации, их достоверного анализа и прогнозирования затрат. Специалистам необходимо освоить анализ и оценивание конкретных факторов, влияющих на характеристики систем и продуктов, вследствие реально существующих и потенциально возможных условий и ограничений. Это привело к появлению области индустриальной науки и практики – ***проектирования и производства сложных систем и продуктов***, в составе общей экономики некоторых предприятий и отраслей. Ее основной задачей являлись анализ, прогнозирование, эффективное управление, распределение ресурсов и экономное использование необходимых капиталовложений в производство ***сложных заказных технических изделий различного назначения***.

Приступая к созданию сложных технических систем и продуктов, заказчики и разработчики, прежде всего, должны ***понять целесообразна ли их разработка*** и оценить возможную эффективность применения готового продукта, оправдаются ли затраты на его производство и использование. Поэтому сложные технические проекты традиционно начинаются с ***анализа и технико-экономического обоснования*** предстоящего проектирования, производства и применения предполагаемого продукта или системы. Заказчику и возмож-

ным потребителям результатов производства необходимо оценивать реальную потребность в его создании и конкурентоспособность, а потенциальному производителю предполагаемого продукта, проводить оценку реализуемости проекта при выделенных условиях и ресурсах, предлагаемых заказчиком [1, 26, 37].

Однако часто разработчики не в состоянии привести заказчику или руководителю проекта достаточно обоснованные доказательства не реальности выполнения выдвигаемых требований к системе и продукту при предложенных ограниченных значениях бюджета и сроков. Руководители конкретных проектов зачастую не в состоянии достаточно обоснованно определять, сколько времени и затрат труда потребуется на каждый этап проектирования и производства системы, и не могут оценивать, насколько успешно выполняется план. Это, как правило, означает, что проект системы с самого начала ***выходит из-под экономического контроля*** и возможна катастрофа с реализацией и завершением создания всей системы в требуемый срок с заданным заказчиком бюджетом и качеством.

На каждом этапе проектирования и производства должен ***проводиться поиск эффективных технических решений реализации системы и продукта***. Их производство зависит от результатов и качества выполнения частных работ и может требовать оперативной корректировки плана. При этом определяющими являются ***организация, стимулирование, контроль состояния и развития проекта***. Для этого необходимо следить за ходом исполнения проекта на всем протяжении его жизненного цикла и сравнивать запланированные и фактические результаты производства. Контроль проектирования и производства является органической функцией управления проектами и должен иметь средства регулирования поведения и применения отдельных специалистов и коллектива производства в целом. ***Объектами управления и контроля при этом являются***: затраты ресурсов на выполнение частных работ и реализацию компонентов продукта (трудоемкость, стоимость, время, материальные ресурсы); графики работ, степень их выполнения, наличие и причины отклонений реализации частных работ от планов, угроза нарушения сроков контракта.

Управление промышленным проектированием и производством – это особый вид рациональной, коллективной деятельности, включающий постановку задач, подготовку решений, планирование, организацию и стимулирование специалистов, контроль хода работ и

использования ресурсов при создании сложных систем и продуктов. Задачи целевого управления такими работами – сводить воедино усилия руководителей, исполнителей – специалистов разной квалификации, подрядчиков и субподрядчиков, добиваясь, чтобы они ***выступали как целевая «команда»***, а не как разрозненная группа функциональных специалистов с индивидуальными целями. Они характеризуются производственно-техническим, организационным и социальным единством [11, 26], которое определяется комплексом средств производства, обладающих технологической взаимосвязью отдельных этапов производственных процессов, в результате которых используемые на предприятии объекты и процессы превращаются в готовую продукцию. В результате должна обеспечиваться концептуальная целостность создания технических систем и высокое качество решения главных задач заказчиков при сбалансированном использовании ресурсов на все функциональные задачи. Рациональная организация предполагает такой способ экономически эффективного соединения всех производственных компонентов в единую взаимосвязанную систему, при которой используется наименьшее количество трудовых ресурсов, предметов труда и средств производства для изготовления требующегося продукта.

Организационное единство предприятия определяется наличием руководства и коллектива, общностью результатов работы – объемом и высоким качеством созданной и реализованной продукции, уровнем рентабельности, размером прибыли, техническими фондами предприятия. Для предприятий определенной сферы деятельности важное значение имеет унификация – применение типовых стандартов, конструктивных и технологических решений, производственной структуры, типовой документации. Это подразумевает упорядоченный комплекс производственных подразделений, организацию управления предприятием и обслуживание работников, их достаточное количество, квалификацию, взаимосвязи и соотношения между ними по размеру занятых оборудования и пропускной способностью при ***проектировании и производстве технической продукции.*** Одной из основных задач при формировании структуры предприятия является ***обеспечение скоординированного функционирования*** всех составляющих производства: подготовительных операций, проектирования, основных производственных процессов, технического обслуживания.

Организационная структура ***управления предприятием*** – упорядоченная совокупность служб, управляющих его деятельностью и взаимосвязями. Структура аппарата управления зависит от многих факторов: типа производства, специализации, объема производства, конструктивной сложности изготавливаемой продукции. Они должны ***обеспечивать рациональное разделение труда*** и определение на этой основе профессионального и квалифицированного состава специалистов, научную организацию и оптимальное обслуживание рабочих мест, улучшение условий труда.

Руководство производственного предприятия должно быть уверено, что требования заказчика полностью поняты и могут быть удовлетворены. Предприятие должно установить общесистемные процедуры для ***управления требованиями и документами***, необходимыми для функционирования системы проектирования и производства, обеспечивающей уверенность, что документы анализируются, при необходимости уточняются и снова утверждаются. ***Управление специалистами*** необходимо для гарантирования, что они являются компетентными для осуществления своей деятельности, на основе соответствующего образования, подготовки, мастерства и опыта при проектировании и производстве заказанной продукции.

Для интеграции усилий специалистов и эффективного использования ресурсов производства ***выделяется руководитель – лидер, управляющий проектированием и производством***. Он формирует политику в области качества, которая определяет стратегические цели, принципиальные направления деятельности и всю идеологию документов системы менеджмента качества. Задача управляющего производством наряду с прямыми воздействиями на подчиненных и координацией их работ – стимулировать и контролировать активность прямых связей между исполнителями частных работ. Руководитель должен установить и управлять процессами, необходимыми для обеспечения уверенности в том, что продукция соответствует требованиям заказчика. В качестве способа внедрения и демонстрации реализации, установленных процессов, ***предприятие должно применять систему менеджмента качества***, основываясь на требованиях международных стандартов.

Для ***эффективного процесса*** изготовления определенного продукта и его реализация в конкретных условиях предприятия необходимо ***создание соответствующей технологии его проектирования и производства***. Технология производства – это методы, техниче-

ские, инструментальные средства и система взаимосвязанных способов изготовления продукции и выполнения установленного вида работ. Технология должна включать весь перечень последовательных операций по превращению исходного замысла проекта, требований заказчика и потребителей, в готовый продукт, с указанием методов, типа и характеристик инструментов, которыми специалисты пользуются на каждом этапе производства.

Организация технологии проектирования и производства – система мер, направленных на рационализацию взаимодействия в пространстве и времени материальных компонентов и специалистов, занятых в процессе производства. Дифференциация коллективного труда предполагает разделение производственного процесса на отдельные части (процессы, операции) и их закрепление за соответствующими подразделениями и специалистами предприятия. В практической деятельности по организации производства, приоритет должен отдаваться тому методу, который обеспечит ***наилучшие экономические и социальные характеристики производственного процесса*** [1, 41].

План проектирования и производства должен отражать ***рациональное сочетание целей, стратегий действий, конкретных процедур, доступных ресурсов и других компонентов,*** необходимых для достижения поставленной основной цели создания системы или продукта с заданным качеством. Планирование производства должно обеспечивать компромисс между требующимися характеристиками создаваемой системы или продукта и ограниченными ресурсами, доступными для их разработки и применения. Предприятие должно определить, спланировать и внедрить процессы измерений, мониторинга, анализа и улучшения производства для обеспечения уверенности в том, что система менеджмента качества, процессы и продукция соответствуют заданным высоким требованиям.

* * *

Представленные выше ***общие индустриальные принципы проектирования и производства сложных технических систем*** в монографии используются и адаптируются для создания ***современных сложных заказных программных продуктов систем управления и обработки информации реального времени.*** Этот вид программных продуктов заказывается, разрабатывается и активно применяется в крупных системах динамического управления объек-

тами и процессами. Для отличия их от создания «свободного программного обеспечения» ниже рассматриваются только сложные ***«заказные»*** системы и программы обработки информации и управления, применяемые в высокоточном технологическом производстве, в авиации, в управлении космическими аппаратами, атомными электростанциями, оборонной техникой. Они являются одними из наиболее сложных компонентов ***интеллектуальных систем особенно высокого качества***, создаваемых человеком. При проектировании и производстве сложных заказных программных продуктов используются основные ***понятия, принципы и процессы современного индустриального производства сложных технических систем***. Такие системы и программные продукты создаются на промышленных предприятиях с использованием принципов современных индустриальных технологий производства сложных технических систем. Такие комплексы программы проектируются и производятся как промышленные изделия по методам, правилам и стандартам, характерным для производства сложной высококачественной технической продукции.

Методология программной инженерии и международные стандарты регламентируют современные методы и средства проектирования и производства, внедрения и применения программных продуктов в составе сложных систем. Организованные коллективы специалистов, применяющие традиционные, регламентированные процессы программирования, верификации, тестирования и сопровождения программных комплексов и их компонентов обеспечивают заказчикам ***стабильные, предсказуемые результаты*** – программные продукты с требуемым качеством обработки технической информации. Основной интеллектуальный труд специалистов вкладывается в разработку функциональных алгоритмов и текстов программ, а также в их интеграцию, испытания, документирование на всех этапах проектирования и производства сложных программных продуктов. Большая сложность информации в таких системах определяет высокие ***затраты квалифицированного интеллектуального труда*** специалистов на комплексы программ, соответствующие требованиям к ним заказчиков и пользователей, необходимого для ***производства и материализации информации в программном продукте***.

Для каждого программного продукта, выполняющего ответственные функции, при определении и реализации требований контрактов и технических заданий заказчиков, должна применяться система

менеджмента качества. Она включает методы и инструментальные средства производства, испытаний и сертификации, обеспечивающие ***высокое качество обработки информации, надежность, безопасность функционирования и применения программных продуктов***. Изложенные ниже методы программной инженерии регламентируют и конкретизируют технологические процессы на обработку информации, а также на контроль качества информации отдельных компонентов и комплексов программ на этапах жизненного цикла.

Значительную сложность для специалистов и заказчиков представляет ***промышленная экономика проектирования и производства эффективных программных продуктов,*** оценивание экономики деятельности предприятий, обеспечение в контрактах рациональное сочетание целей, стратегий и использование доступных ресурсов. Заказные программные продукты должны проходить экономическое обоснование и прогнозирование выбора и применения комплексной системы качества, методологии и инструментальных средств, гарантирующих требуемое качество, надежность и безопасность функционирования программного продукта. ***Предсказуемые высокие экономические характеристик производства***, и достигаемого качества программного продукта, должны сопровождать весь жизненный цикл сложных комплексов программ.

Развитие экономики этой области индустрии ***связано с большими профессиональными и психологическими трудностями***, типичными для новых разделов современной науки и техники, проявляющимися на стыке различных областей методов и знаний. В данном случае особенности состоят в том, что ***менеджеры и разработчики*** сложных заказных программных продуктов, как правило, не знают даже основ промышленного производства и экономики сложной технической продукции. ***Заказчики и экономисты,*** зачастую, не представляют сущность и детальные свойства объектов разработки – программных продуктов, особенностей экономики технологических процессов их проектирования, производства и применения. Широкий спектр технических характеристик таких объектов, количественных и качественных факторов, которые с различных сторон характеризуют содержание компонентов и комплексов программ, и невысокая достоверность экономической оценки их значений, определяют значительную ***трудность при попытке описать и измерить*** свойства, количество и качество сложных про-

граммных продуктов ***для их экономической оценки и прогнозирования характеристик***.

После завершения производства и испытаний компоненты и комплексы ***отчуждаются*** от коллектива их создателей. Они могут применяться, эксплуатироваться, эпизодически модифицироваться и корректироваться. Это приводит к необходимости оформлять промышленную, достаточно полную ***технологическую документацию***, позволяющую различным квалифицированным специалистами применять и развивать заказные программные продукты. Понятия, измеряемые экономические характеристики, трудоемкость организованного проектирования и производства программных продуктов должны учитываться как ***ориентиры*** в процессе оценивания, сравнения и прогнозирования сложности и количества информации в заказных системах.

Определяющую роль в сложных технических системах играет количество ***интеллектуального труда специалистов***, необходимого для ***создания и материализации информации в программах***. При этом рекомендуется принимать во внимание тот факт, что сложные программные продукты, кроме функциональной, интеллектуальной части, определяющейся основными алгоритмами системы, содержат значительную часть программ, выполняющих стандартные, вспомогательные функции с относительно малой информационной сложностью и трудоемкостью их производства. Такие программы необходимо учитывать при формализации количества и сложности информации в заказных системах, для оценивания размеров и бюджетов создания программных продуктов ***при заключении контрактов*** между заказчиками и разработчиками на их проектирование и производство.

Понятия, экономические характеристики и измеряемую трудоемкость организованного проектирования и производства программных продуктов необходимо использовать как основу при оценивании, сравнении и ***прогнозировании сложности и количества информации в заказных системах***. Это необходимо при системном проектировании и сопоставлении эффективности вариантов предлагаемых к реализации программных продуктов.

Для эффективного проектирования и производства программных продуктов необходимо обучение и подготовка квалифицированных коллективов руководителей и специалистов, освоение современных методов анализа, оценивания и прогнозирования процессов и необходимых ресурсов при производстве комплексов про-

грамм. Следует выделять и обучать специалистов, ***ответственных*** за соблюдение экономически эффективной промышленной технологии создания и сертификации программных продуктов. Руководителям и специалистам необходимо освоить современный комплекс ***экономически эффективных технологических методов и стандартов*** промышленного создания и развития сложных, заказных программных комплексов и баз данных гарантированного высокого качества.

Ключевой задачей на всех стадиях проектирования и производства заказных программных продуктов является ***диагностика, выявление и устранение дефектов и ошибок***. Понимание, прогнозирование и достижение высокого качества комплексов программ однозначно связано с контролем, регистрацией и ликвидацией всех возможных аномалий их функционирования. Следует учитывать, что затраты на обнаружение и устранение дефектов быстро возрастают при приближении к завершающим этапам испытаний и сертификации сложных программных продуктов, поэтому прогнозирование и ликвидация ошибок должна быть ***доминирующей задачей обучения*** всех руководителей и специалистов каждого заказного проекта.

При изложении производственных процессов ниже активно используются международные стандарты, многие из которых адаптированы на русском языке (Приложение 1). В большинстве глав монографии приводятся рисунки с кратким содержанием рассматриваемых задач и процессов, которые иллюстрируют содержание монографии. Их полезно использовать для подготовки иллюстраций ***к курсу лекций при обучении проектированию и производству сложных заказных программных продуктов высокого качества.***

Часть 1

ПРОЕКТИРОВАНИЕ ЗАКАЗНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Глава 1.1

СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ КОМПЛЕКСОВ ПРОГРАММ

Принципы системного проектирования комплексов программ

Основная цель системного проектирования программных комплексов – подготовить, обосновать и согласовать замыслы и решения заказчика (потребителя) и разработчика (поставщика) о необходимости, направлениях и концепции создания или модернизации существующего ПК и изменениях его качества. Методы и средства системного проектирования должны подготавливать эффективную технологическую базу для обеспечения всего жизненного цикла ПК требуемого качества. Характеристики комплексов программ должны анализироваться и формулироваться в начале их жизненного цикла и определять эффективность всех последующих процессов. Результатом этих работ должны быть ***системный проект, техническое задание и контракт*** на продолжение разработки ПК или решение о её нецелесообразности и прекращении – рис.1.1.

Непредусмотренные при системном проектировании ситуации и возможные дефекты программ являются потенциальными источниками отказов и аварий при применении ряда систем. Массовая практика, когда ***заказчик не может сформулировать*** четкие требования к функциям и безопасности ПК, а ***разработчик не понимает***, что нужно заказчику, приводит к длительному процессу разработки проектов с множеством дефектов и ошибок, на устранение которых расходуются большие ресурсы. В результате многие системы не соответствовали исходному назначению и первоначальным спецификациям, не укладывались в графики и бюджет разработки.

Системное проектирование комплексов программ включает:

- принципы системного проектирования комплексов программ;
  - результаты выполненных системных исследований и разработок комплекса программ;
  - концепцию создаваемого комплекса программ на естественном языке данной предметной области;
  - адекватно описаны цели и объект проектирования ПК;
  - предварительные спецификации требований к комплексу программ, его программным и информационным компонентам;
  - стратегическое планирование проектирования ПК:
  - рациональное использование ресурсов в процессе создания сложных ПК гарантированного качества;
  - предварительное технико-экономическое обоснование проекта, приближенные значения трудоемкости и длительности всей разработки ПК;
- структурное проектирование программного комплекса;
  - спецификация требований к предварительной архитектуре комплекса программ;
  - многоуровневое, иерархическое построение программного комплекса;
  - стандартизация структуры межмодульных интерфейсов компонентов по передачам управления и по информации;
  - унифицированные правила структурного построения и оформления спецификаций требований компонентов.

Рис. 1.1

Поэтому значительно возросла необходимость освоения всех современных методов и методик ***предупреждения системных дефектов проектирования***. Многие ошибки, обусловленные неопределенностью или некорректностью технических заданий и спецификаций, могут и должны быть выявлены на ранних стадиях системного проектирования, что способствует его ускорению и повышению качества. Обширной практикой доказано, что обнаружение и устранение ошибок и дефектов в комплексах программ на начальных этапах системного ***проектирования в десятки и сотни раз быстрее и дешевле, чем в процессе завершения разработки и испытаний***.

Для управления проектом системы, прежде всего, должен быть ***адекватно описаны цели и объект проектирования***. Для сложных систем формализация и детализация характеристик объекта разработки происходит одновременно с процессом его проектирования.

Последовательно уточняются архитектура объекта, основные функции и их характеристики, требующиеся показатели качества функционирования и методы решения задач. Все эти данные отражаются в концепции, техническом задании, спецификации требований и описании проекта, которые детализируются и конкретизируются по мере развития проекта. Это определяет принципиальную особенность планирования проектов сложных систем, состоящую в наличии влияния на план ***изменяющихся значений и достоверности знаний*** заказчика и разработчиков о требуемых характеристиках объекта разработки. С этим связана необходимость итерационного уточнения планов на всех этапах проектирования, разработки и совершенствования систем [1,13].

План проекта должен отражать рациональное сочетание целей, стратегий действий, конкретных процедур и доступных ресурсов, необходимых для достижения поставленной ***основной цели проекта с заданным качеством***. Планирование проектов должно ***обеспечивать компромисс*** между требующимися характеристиками создаваемой системы и ограниченными ресурсами, необходимыми на ее разработку и применение. По мере уточнения исходных данных об объекте разработки, внешней среде применения и ресурсах, в процессе системного анализа и проектирования возрастает достоверность планирования.

В системном проекте должны быть обобщены и отражены следующие основные ***результаты выполненных системных исследований и разработок*** (см.рис. 1.1):

- обобщенный анализ проведенного обследования объекта информатизации, функций существующей системы, качества ее основных программных компонентов и базы данных;
- совокупность предварительных исходных требований к функциям и характеристикам комплекса программ;
- оценки имеющихся и потенциально доступных ресурсов (финансовых, вычислительных средств, специалистов) для обеспечения всего жизненного цикла и требуемого качества проекта комплекса программ;
- результаты предварительного анализа возможной архитектуры комплекса программ на основе моделей и прототипов аналогичных систем, позволяющие наметить планы разработки и всего жизненного цикла проекта ПК;

- цели, задачи и функции предполагаемой новой или модернизированной системы, обобщенные в ***концепции*** создания соответствующего программного средства;
- проекты жизненного цикла, гарантирования требуемого качества ПК, защиты и обеспечения безопасности его функционирования;
- результаты технико-экономического обоснования целесообразности и основных направлений продолжения проектирования ПК;
- результаты анализа существующей и возможной инструментальной среды разработки, а также системы обеспечения качества, перспективы их развития и совершенствования;
- предварительный план организации работ, требования к составу и квалификации специалистов для выполнения проекта и всего жизненного цикла ПК;
- формализованное ***техническое задание и спецификации требований к ПК***, позволяющие заключить контракт между разработчиком и заказчиком на финансирование и продолжение детального проектирования и/или на весь жизненный цикл ПК.

При создании сложных ПК важно учитывать, что только заказчик и потенциальный пользователь системы вправе и способен корректно формулировать требования и впоследствии судить, насколько успешно проведена разработка соответствующего программного продукта. Аналитики-консультанты совместно с потенциальными разработчиками и заказчиком или пользователями должны проводить анализ прикладной области и объекта информатизации, разрабатывать стратегию разработки и технико-экономическое обоснование реализуемости выдвигаемых требований.

Эта деятельность требует специальной организации специалистов наивысшей квалификации и тесной совместной работы представителей заказчика и разработчика. Они должны подготовить исходные данные и документы, в которых содержатся предварительные требования и пожелания к функциональным и конструктивным характеристикам качества программного комплекса. Далее ими должна проводиться сложная работа по предварительному упорядочению, селекции, обобщению и ***ранжированию приоритетов требований*** для их реализации в проекте. Наличие обычно ряда неформализованных, неструктурированных и противоречивых тре-

бований заказчика и разработчика требует их совместной обработки, согласования и корректировки.

Функциональные требования заказчика к процессам и результатам обработки информации необходимо скоординировать с конструктивными требованиями и возможностями их эффективной реализации разработчиками в ***спецификациях требований к комплексу программ*** и его программным и информационным компонентам. Должна быть предусмотрена корректировка, конкретизация и развитие совокупности предварительных требований в процессе системного проектирования и в дальнейшем по мере реализации проекта при тесном взаимодействии заказчика и разработчика. Для крупных проектов ПК целесообразно использовать специальный инструментарий и хранилище решений в процессе отработки требований, которые следует учесть в системном проекте и техническом задании, а также применять для контроля их реализации.

Предварительный анализ и моделирование процессов обработки данных при системном проектировании должны проходить этапы от простого установления базовых отношений между понятиями, через определение интерфейсов доступа и атрибутов, к проекту модели состояний и взаимодействий между реальными объектами, компонентами и процессами ПК. Эти модели должны служить базой при разработке схем потоков управления и данных, описывающих процессы их обработки, а впоследствии интегрироваться с отработанными моделями процессов для комплексного исследования функционирования прототипов – пилотных проектов ПК в целом.

При построении формализованного описания системы, выполняемом её разработчиком, принципиальными являются ***два организационных момента***: специалисты – заказчики или пользователи создаваемой системы должны активно участвовать в процессе анализа и реализации её описания; каждый шаг описания должен обязательно документироваться. Наглядными и удобными в работе являются ***графические представления описаний проектных решений***, которые позволяют создавать прототипы ПК. Они обеспечивают эффективную визуализацию и обратную связь между разработчиком и потенциальным пользователем с целью оценки реализации требований, корректировки функций и качества компонентов, а также форм пользовательского интерфейса. Схемы потоков данных, потоков управления, сущность-связь и другие – составляют комплекс

удобных и гибких графических методов и средств описания систем, ***облегчающих взаимопонимание между разработчиками и заказчиками*** на разных уровнях детализации функций, качества и архитектуры ПК.

Она является базовым исходным документом, согласуемым с заказчиком для создания комплекса программ. На основе этого описания формируется предварительное техническое задание на систему и её основные компоненты. При использовании формализованных методов разработки программных средств текстуальное описание системы подлежит переводу на соответствующий, возможно, графический язык. Наряду с разработчиками, специалисты-заказчики или пользователи создаваемой концепции ПК должны активно участвовать в процессе анализа и реализации её описания.

Стратегическое планирование проекта должно содержать долгосрочные цели развития ПК определенного функционального назначения. На базе требований к ПК и первичных планов появляется возможность оценить объем подлежащих разработке компонентов программ и баз данных, а также некоторые дополнительные характеристики возможного объекта и среды разработки. По этим данным руководителем разработки и заказчиком принимается решение о целесообразности продолжения проектирования и осуществляется стратегическое планирование проекта, которое формализуется в техническом задании на ПК. Эти данные позволяют принимать решения по корректировке требований к ПК, по изменению среды разработки или состава коллектива специалистов. Таким образом, последовательное прогнозирование, планирование и системное управление проектом ***обеспечивают рациональное использование ресурсов*** в процессе создания сложных ПК гарантированного качества [4, 35].

Достоверность планов и прогнозов определяется точностью сведений об объекте разработки, характеристиках технологической среды и прототипов, принятых за основу при планировании. Таким образом, ***производится технико-экономическое обоснование проекта***, определяются приближенные значения трудоемкости и длительности всей разработки ПК, а также число необходимых специалистов. Вследствие творческого характера большинства работ на этом этапе невозможно составить жесткий план их выполнения. Помочь может типовой перечень частных работ, представленный в

стандартах и ориентировочный график, иллюстрирующий их взаимосвязь.

Проведенные таким образом оценки проекта позволяют осуществить ***предварительный выбор основных методов и инструментальных средств*** для проведения последующего детального и рабочего проектирования и поддержки всего ЖЦ ПК. При этом должно активно использоваться ***моделирование и тестирование корректности системных решений***. Благодаря высокому качеству проработки и документирования системного проекта, создается основа для снижения трудоемкости тестирования, испытаний, а также сопровождения и модификации ПК. В процессе системного проектирования должны предварительно определяться состав и структура основных технологических и эксплуатационных документов для поддержки всего ЖЦ ПК. Эти документы должны обеспечивать реализацию процессов жизненного цикла ПК, планирования и управления, регистрировать выполнение требуемых действий, формализовать систему качества. При этом следует подготовить первоначальные ***требования к документации и обеспечить их реализацию***, которая должна быть однозначной – написана в стандартизированных терминах, которые допускают только единственную интерпретацию, уточняемую, если необходимо, соответствующими комментариями. Если заказчик удовлетворен результатами системного проектирования, то возможно оформление ***акта*** завершения работ и утверждение системного проекта комплекса программ с требуемыми характеристиками качества новой или модернизированной системы, а также ***контракта (договора)*** на детальное проектирование или на весь жизненный цикл комплекса программ.

Структурное проектирование сложных программных комплексов

Основные принципы и правила структурирования ПК можно объединить в группы, которые отражают (см. рис. 1.1):

- стандартизированную структуру целостного построения ПК и/или БД определенного класса;
- унифицированные правила структурного построения функциональных программных компонентов и модулей;
- стандартизированную структуру базы данных, обрабатываемых программами;

- унифицированные правила структурного построения информационных модулей, заполняющих базу данных;
- унифицированные правила организации и структурного построения межмодульных интерфейсов программных компонентов;
- унифицированные правила внешнего интерфейса и взаимодействия компонентов ПК и БД с внешней средой, с операционной системой и другими типовыми средствами организации вычислительного процесса, защиты и контроля системы [8, 19].

Функции должны детализироваться ***сверху вниз*** в виде иерархической структуры таким образом, чтобы процедуры сбора, хранения и переработки информации, рассматриваемые сначала как нечто единое целое, расчленялись на отдельные элементы данных, компонентов и действия, совершаемые над этими данными. Структурный анализ, исходя из функционального описания системы в целом, позволяет разделить её на функциональные части, выделить функциональные описания отдельных частей, исследовать в них информационные потоки и формализовать структуры данных. Когда различные проектные решения прочно взаимосвязаны, то бывает полезно, чтобы всеми ими сразу занимались в одно и то же время одни и те же люди. Прежде всего, необходимо пытаться изолировать функции, которые менее всего связаны с другими. Затем рассматривать раздельно функции, учитывая ***имеющие отношения между собой*** и детали связанных проблем.

Спецификация требований к предварительной архитектуре комплекса программ формируется в процессе детализации и уточнения спецификации требований к характеристикам ПК в результате проверки последних на непротиворечивость и полноту. Задача спецификации требований архитектуры состоит в том, чтобы представить достаточно ясное и удобное общее описание внешнего поведения системы и свойств, а также её внутренней структуры и механизмов функционирования.

Описания проектных решений должны содержать первичные спецификации крупных функциональных компонентов, подлежащих разработке в детальном проекте создаваемой системы, и спецификаций используемых готовых компонентов, состав которых определяется ***при декомпозиции*** общей структуры системы. В ***требованиях спецификации к системной архитектуре*** комплекса программ должно обеспечиваться:

- соответствие функций и структуры ПК аппаратной и операционной среде, их ресурсам и интерфейсам;
- совместимость ПК с другими системами по источникам и потребителям информации;
- соответствие стандартам структурного построения и интерфейсов комплекса программ, функциональных компонентов и модулей;
- предварительная организация информационного обеспечения и структура базы данных;
- состав, структура и способы обмена данными между функциональными компонентами и внешней средой;
- временной регламент и предварительные характеристики процессов реализации функций, интенсивность и объемы потоков информации базы данных;
- контроль, хранение, обновление, защита и восстановление программ и данных;
- стандартизированные, предварительные требования к составу и содержанию технологической документации.

Структурное проектирование программных комплексов ***основано на модульном принципе***. Многоуровневое, иерархическое построение сложных программных комплексов позволяет ограничивать и локализовать на каждом из уровней соответствующие ему компоненты. Нижнему иерархическому уровню представления программ соответствуют программные и информационные модули (модули данных). Эти компоненты объединяются в группы программ определенного функционального назначения с автономной целевой задачей. Несколько групп функциональных программ образует комплекс программ. В особо сложных случаях возможно создание программного комплекса из нескольких взаимодействующих комплексов. Всем иерархическим системам (в частности, ПК) ***присущ ряд общих свойств***, важнейшими из которых являются:

- вертикальная соподчиненность, заключающаяся в последовательном упорядоченном расположении взаимодействующих компонентов, составляющих ПК;
- право вмешательства и приоритетного воздействия сверху вниз на компоненты нижних уровней;
- взаимозависимость действий компонентов верхних уровней от реакций на воздействия и от функционирования компонентов

нижних уровней, информация о которых передается верхним уровням.

В результате в иерархических структурах ПК образуются ***два потока взаимодействий*** между компонентами разных уровней: ***сверху вниз*** – координирующие и управляющие воздействия верхних уровней и ***снизу вверх*** – информация о состоянии и реализации предписанных сверху функций компонентами нижних уровней. Степень автономности компонентов и интенсивность координирующих воздействий устанавливаются в результате ***компромисса при выделении числа и размеров иерархических уровней***. Взаимодействие компонентов в пределах уровня целесообразно максимально ограничивать, что позволяет упростить общее координирование компонентов и проводить его только по вертикали.

Менее наглядными являются ***иерархия данных, обрабатываемых ПК***, и их взаимодействие с программными компонентами. Функциональная иерархия данных отражается ***расстоянием*** между расчетом или изменением переменной и её использованием, или условной длительностью хранения неизменяемых значений переменной. Взаимодействие двух программных модулей может осуществляться так, что некоторые переменные используются только этими модулями. Такие обменные переменные имеют более широкую область применения и должны храниться все время, пока не будут вызваны взаимодействующие модули. Ряд переменных и массивов используется многими модулями и группами программ в комплексе – это глобальные переменные. Они характеризуются наиболее широким использованием и соответствуют высшему иерархическому уровню среди данных.

Анализ концепции, требований технического задания и технико-экономических оценок должен позволять выполнить ***предварительное структурное проектирование ПК и оценку вычислительных ресурсов*** необходимых для решения основных функциональных задач. Повышению эффективности структурирования могут значительно способствовать заимствование из предыдущих проектов спецификаций прототипов, версий и отдельных компонентов. Для обеспечения системного проектирования на этом этапе большое значение имеют графические методы визуализации технических решений и логического контроля проекта.

Характеристики внешней среды применения ПК и особенности реализующего компьютера в значительной степени определяют архи-

тектуру и структуру применяемой операционной системы, средств контроля и организации вычислительного процесса. При разработке версий ПК для некоторой прикладной области целесообразно выбирать и унифицировать внешний интерфейс и операционную систему. Это обеспечивает многократное использование одних и тех же организующих программ, дисциплинирует структурное построение версий ПК и способствует унификации межмодульного интерфейса.

Учет возможности изменений – это принцип, который более всего отличает программное средство от большинства других типов промышленных продуктов. Во многих случаях структура КП разрабатывается, когда требования заказчика к нему осознаны не полностью. Позже, при детализации и после поставки, ПК должно развиваться на основе откликов заказчика и пользователей, поскольку обнаруживаются новые требования, а старые уточняются. В дополнение к этому, программный продукт часто встраивается в среду, которая воздействует на него, и это воздействие генерирует новые требования, которые не были изначально известны.

Повторная применимость – является еще одним качеством структуры программного продукта, на которое заметно воздействует принцип возможности изменений. Компонент является многократно применимым, если он может быть непосредственно использован для производства нового продукта или версии. На практике, компонент может претерпеть некоторые изменения прежде, чем будет использован повторно. Отсюда, многократное использование можно рассматривать как эволюционность системной структуры ПС на уровне компонентов. Если можно предвидеть изменения контекста, в который будет встроен программный компонент, то и компонент можно спроектировать таким образом, что изменения будут им учтены.

Большинство промышленных процессов являются ***модульными*** и составлены из комплексов работ, которые комбинируются простыми способами (последовательными или перекрывающимися) для достижения требуемого результата. Главное преимущество модульности заключается в том, что она позволяет применять принцип разделения задач на ***двух этапах***:

- при работе с элементами каждого модуля отдельно (игнорируя элементы других модулей);
- при работе с общими характеристиками групп модулей и отношениями между ними с целью объединить, их в конкретный, более крупный и сложный компонент.

Если данные этапы выполняются в последовательности, предусматривающей сначала концентрацию процессов на модулях, а затем – их объединение, то система проектируется ***снизу вверх***. Если сначала систему разбивают на модули, а потом работают над их индивидуальным проектированием, то это – проектирование ***сверху вниз***.

При структурном построении комплексов программ важное значение имеет размер и сложность компонентов для каждого уровня иерархии и соответственно число иерархических уровней для крупных ПС. По принципам построения, языку описания, размеру и другим характеристикам компонентов в структуре ПС ***можно выделить иерархические уровни***:

- программных модулей, оформляемых как законченные компоненты текста программ;
- функциональных групп (компонентов) или пакетов программ;
- комплексов программ, оформляемых как законченные программные продукты определенного целевого назначения.

С повышением иерархического уровня увеличивается размер текста программ, реализующих компоненты этого уровня и количество обрабатываемых переменных. Одновременно совокупности команд все более специализируется и снижается возможность повторного применения компонентов в различных комбинациях для решения аналогичных задач.

Программные модули решают относительно небольшие функциональные задачи, и каждый реализуется 10-100 операторами языка программирования. Каждый модуль может использовать на входе около десятка типов переменных. Если для решения небольшой функциональной задачи требуется более 100 операторов, то обычно целесообразно проводить декомпозицию задачи на несколько более простых модулей.

Функциональные группы программ (компоненты) формируются на базе нескольких или десятков модулей и решают достаточно сложные автономные задачи. На их реализацию целесообразно использовать до десятка тысяч строк текста программы. Соответственно возрастают число используемых типов переменных и разнообразие выходных данных. При этом быстро растет число типов переменных, обрабатываемых модулями и локализующихся в пределах одного или нескольких модулей.

Комплексы программ – программные продукты создаются для решения сложных задач управления и обработки информации. В

комплексы объединяются несколько или десятки функциональных групп программ для решения общей целевой задачи системы. Размеры ПК зачастую исчисляются сотнями модулей, десятками и сотнями тысяч операторов. Встречаются ПК, содержащие до двух-трех десятков структурных иерархических уровней, построенных из модулей.

Программные модули для их многократного использования должны базироваться на ***унифицированных правилах структурного построения***, оформления спецификаций требований и описаний текстов программ и комментариев. Кроме того, целесообразно для каждого проекта директивно ограничивать размеры модулей по числу строк текста с учетом языка программирования, например, 30-ю или 50-ю операторами. При их применении целесообразно выделять типовые ассоциации операторов и ограничения их использования, а также, вводить правила описания текстов программ, комментариев, данных и заголовков модулей, ограничения их размеров и сложности. Эти правила наиболее полно должны соблюдаться при разработке основной массы функциональных программ, подлежащих повторному использованию в модифицируемых версиях программных продуктов. Компоненты организации вычислительного процесса, контроля функционирования, ввода-вывода и некоторые другие могут иметь отличия в правилах структурного построения модулей.

Для обеспечения управляемой модификации и развития конфигураций версий программного продукта важно ***стандартизировать структуру базы данных***, в которой накапливается и хранится исходная, промежуточная и результирующая информация в процессе функционирования КП. Основными компонентами этой структуры являются информационные модули или пакеты данных. В них также целесообразно использовать типовые структуры, ориентированные на эффективную обработку данных в конкретной проблемной области. Объединение информационных модулей позволяет создавать более сложные структуры данных определенного целевого назначения. Иерархия связей между этими компонентами в некоторой степени соответствует процессу обработки и потокам данных и относительно слабо связана со структурой программных компонентов в комплексе.

Особое значение для качества модулей и компонентов крупных ПК имеет ***стандартизация структуры межмодульных интерфейсов по передачам управления и по информации***. Эти правила формируются на базе описаний языков программирования или оформляются на основе правил структурного построения программ и базы

данных конкретных проектов ПК. В последнем случае соглашения о связях конкретизируются в макрокомандах межмодульного взаимодействия. ***Структурное проектирование*** сложных комплексов программ активно развивается на основе концепции и стандартов открытых систем. Применение стандартов открытых систем следует начинать при создании ***архитектуры исходных модулей*** мобильных ПК и БД, а далее использоваться при всех процессах ЖЦ. Во всех случаях создание архитектуры модулей и компонентов современных сложных систем целесообразно вести с использованием профилей международных стандартов, значительная часть которых обеспечивает мобильность и возможность повторного использования готовых программных средств и баз данных.

Основными целями создания и применения унификации интерфейсов является повышение общей экономической эффективности разработки и функционирования систем, а также логической и технической совместимости их компонентов, обеспечение мобильности и повторного применения готовых программ и данных. Они реализуют спецификации на интерфейсы, процессы и форматы данных, достаточные для того, чтобы обеспечить:

- возможность расширения ПК, а также переноса (мобильность) программных компонентов и систем, разработанных должным образом, с минимальными изменениями на широкий диапазон аппаратных и операционных платформ;
- совместную работу с другими программными продуктами и системами на локальных и удаленных платформах;
- взаимодействие с пользователями в стиле, облегчающем последним, переход от системы к системе (мобильность пользователей).

Проектирование и производство заказных технических систем и программных продуктов регламентируют ***четыре крупные комплексы международных стандартов***:

- **Стандарт ГОСТ Р ИСО/МЭК 12207.2007 – Системная и программная инженерия, процессы жизненного цикла систем и программных комплексов;**
- **CMMI – Система и модель оценки зрелости, управление проектами программных средств;**
- **ГОСТ Р ИСО/МЭК 9000:2000 – Стандарты менеджмента (административного управления) качеством систем;**
- **Стандарт ISO 19759:2005 – SWEBOK, Совокупность знаний о разработке программных средств. Руководство.**

Системное проектирование целесообразно завершать выбором и адаптацией компонентов из этих стандартов, и подготовкой комплекса Руководств и рабочих инструкций для коллектива специалистов – разработчиков конкретной системы и программного продукта с учетом их характеристик и внешней среды.

Системная и программная инженерия, процессы жизненного цикла систем и программных комплексов

Стандарт **ГОСТ Р ИСО/МЭК 12207.2007** – ***предоставляет широкую совокупность процессов***, облегчающих связи между заказчиками, производителями, приобретающими сторонами, поставщиками и другими правообладателями в ***течение жизненного цикла систем и программных продуктов*** – рис. 1.2. Он разработан для организаций, проектирующих, разрабатывающих, приобретающих системы и программные продукты, а также для менеджеров (в том числе, менеджеров по качеству) и пользователей программных продуктов. Стандарт предназначен для использования при двусторонних отношениях ***заказчик – поставщик*** и может применяться также в случае, когда обе стороны принадлежат одной и той же организации. Такие отношения могут варьироваться от неформального соглашения вплоть до официального контракта.

В стандарте ***организация – это группа лиц*** с определенными обязанностями и полномочиями, объединенная для реализации некоторых конкретных процессов. Процессы в стандарте составляют ***полную совокупность*** для охвата различных функций проектирования и производства систем или программных продуктов.

Организация (малая или крупная) в зависимости от её деловых целей или стратегии приобретения и построения продукта, может выбрать подходящую совокупность процессов, также связанных с ними действий и задач, для выполнения этих целей.

Организация может выполнять один или несколько процессов. При условии контракта или применения стандарта организация не обязательно должна выполнять процесс приобретения или поставки. Процесс может выполняться одной или несколькими организациями. В стандарте процессы, их действия и задачи располагаются ***в виде упорядоченной последовательности***, подходящей для реализации пояснений конкретного проекта. Эти последовательности не предполагают и не устанавливают какой-либо зависимости от времени.

Группы процессов жизненного цикла систем и комплексов программ

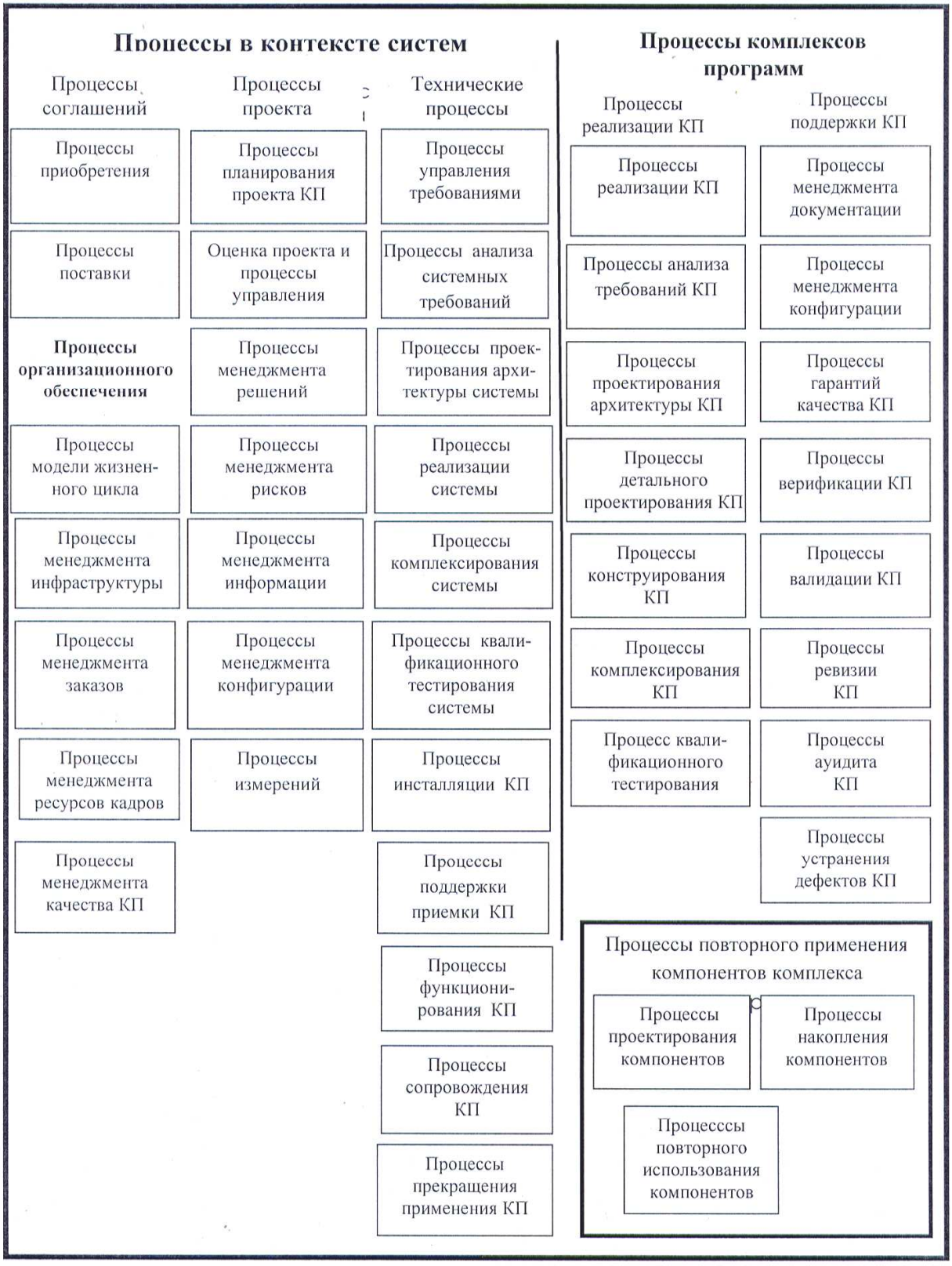

Рис. 1.2.

Из-за невозможности достичь единого мнения или применить универсальную, развернутую во времени последовательность процессов, пользователь стандарта может самостоятельно выбрать и назначить процессы, виды деятельности и задачи, как наиболее подходящие и эффективные для определенного проекта. Настоящий стандарт способствует выполнению итераций между действиями и рекурсий в пределах отдельных действий для того, чтобы нейтрализовать нежелательное влияние любой последовательности действий и задач. Организации, применяющие стандарт, ответственны за выбор модели жизненного цикла для проекта и отображения процессов, действий и задач в такой модели.

В стандарте устанавливается ***общая структура работ для жизненного цикла систем и для программных комплексов***. Процессы ***в контексте системы*** делятся на три группы: процессы соглашений, проекта системы и технические процессы (всего 23 процесса). Раздел ***процессов комплексов программ*** включает две группы из 18 процессов: процессы реализации и процессы поддержки комплексов. По названиям они в значительной степени подобны процессам в стандарте **ISO 12207:1995**, но различаются по содержанию. Жизненный цикл начинается от замысла или потребности, которая может быть удовлетворена полностью или частично системой или программным комплексом, и завершается прекращением его применения. Такая архитектура создается совокупностью процессов и взаимосвязями между этими процессами. Определение процессов жизненного цикла основываются на двух базовых принципах: связности и ответственности. ***Связность:*** процессы жизненного цикла являются связными и соединяются оптимальным образом, считающимся практичным и выполнимым. ***Ответственность:*** процесс может передаваться под ответственность какой-либо организации в жизненном цикле системы или программного комплекса.

Каждый процесс настоящего стандарта ***описывается в терминах следующих атрибутов***:

- наименование передает область применения процесса, как целого;
- цель описывает конечные задачи выполнения процесса;
- выходы представляют собой наблюдаемые результаты, ожидаемые при успешном выполнении процесса;
- деятельность является перечнем действий, используемых

для достижения выходов;

- задачи представляют собой требования, рекомендации или допустимые действия, предназначенные для поддержки достижения выходов процесса.

Атрибуты характеризуют специфику каждого процесса. Когда реализуемый процесс соответствует этим атрибутам, то специально определенная цель процесса и его результаты достигаются посредством реализации определенной деятельности. В дополнения к этим базовым атрибутам процессы могут характеризоваться другими атрибутами, общими для всех процессов. Каждый процесс стандарта удовлетворяет описанным критериям. С целью четкого описания процессы иногда подвергают декомпозиции на более мелкие части.

Задачи выражаются в форме требования, рекомендации или допустимого действия, предназначенных для поддержки достижения выходов процесса. Для этой цели в стандарте используются конкретные вспомогательные глаголы (должен, следует и может), чтобы подчеркнуть различие между разными формами задач. Глагол «должен» используется для выражения условия, требуемого для соответствия, «следует» – для выражения рекомендации среди других возможностей и «может» – для того, чтобы отразить курс допустимых действий в пределах ограничений настоящего стандарта.

Процесс жизни любой ***системы или программного продукта*** может быть описан посредством ***модели жизненного цикла, состоящей из стадий***. Модели могут использоваться для представления всего жизненного цикла от замысла до прекращения применения или для представления части жизненного цикла, соответствующей текущему проекту. Модель жизненного цикла представляется в виде последовательности стадий, которые могут перекрываться и (или) повторятся циклически в соответствии с областью применения, размером, сложностью, потребностью в изменениях и возможностями. Каждая стадия описывается формулировкой цели и выходов. Процессы и действия жизненного цикла отбираются и исполняются на этих стадиях для полного удовлетворения цели и результатов этой стадии. Организации могут использовать различные стадии в пределах жизненного цикла. Однако каждая стадия реализуется организацией, ответственной за эту стадию, с надлежащим рассмотрением информации, имеющейся в планах

жизненного цикла и решениях, принятых на предшествующих стадиях. Аналогичным образом, организация, ответственная за текущую стадию, ведет записи принятых решений и записи допущений, относящихся к последующим стадиям в данном жизненном цикле. Настоящий стандарт не требует использования какой-либо конкретной модели жизненного цикла. Однако он требует, чтобы в каждом проекте определялась подходящая модель жизненного цикла, предпочтительно та, которая уже определялась организацией для применения в предыдущих проектах.

В стандарте не детализируются процессы жизненного цикла в терминах методов или процедур, необходимых для удовлетворения требований и достижения результатов процесса. Стандарт может требовать разработку документов подобного класса или типа; например, различных планов. Однако, он не предусматривает, чтобы такие документы разрабатывались или комплектовались раздельно или каким-то образом объединялись. Эти решения остаются за пользователем стандарта. Стандарт не устанавливает конкретную модель жизненного цикла системы или программных продуктов, разработку методологии, методов, моделей или технических приемов.

Предполагается, что любой проект проводится в пределах контекста организации. Это важно, так как системный или программный проект зависит от различных результатов, производимых деловыми процессами организации, например, от наемных работников для укомплектования штатного персонала проекта и средств обеспечения проекта. Для этой цели настоящий стандарт предоставляет совокупность процессов ***«организационного обеспечения проекта»***.

Современные организации, ведущие свою деятельность в области сложных программ, стремятся разрабатывать устойчивую совокупность процессов жизненного цикла программных комплексов, которые применяются по нескольку раз для программных проектов в деловой сфере. Поэтому, стандарт предназначен для внедрения либо на уровне организации, либо на уровне проекта. Организации следует внедрить стандарт и дополнить его соответствующими процедурами, практическими рекомендациями, инструментарием и политиками. Программный проект организации обычно следует согласовывать в большей степени с процессами ор-

ганизации, чем согласовывать непосредственно с настоящим стандартом.

Управление проектами программных комплексов в системе CMMI

Назначение методологии **CMMI** – ***системы и модели оценки зрелости*** – состоят в предоставлении необходимых общих рекомендаций и инструкций предприятиям, производящим программные продукты. В выбор стратегии совершенствования качества процессов и продуктов, путем анализа степени их производственной зрелости и оценивания факторов, в наибольшей степени влияющих на качество ЖЦ ПК, а также посредством выделения процессов, требующих модернизации. Для достижения устойчивых результатов в процессе развития технологии и организации управления жизненным циклом ПК рекомендуется использовать эволюционный путь, который позволяет совершенствовать и постепенно повышать качество процессов и продуктов, вскрывать преимущества и недостатки предприятия. В методологии CMMI ***выделены пять уровней зрелости***, раскрываемые в этом стандарте де-факто. ***Уровни зрелости характеризуются*** степенью формализации, адекватностью измерения и документирования процессов и продуктов ЖЦ ПК, широтой применения стандартов и инструментальных средств автоматизации работ, наличием и полнотой реализации функций системой обеспечения качества технологических процессов и их результатов.

Описание процессов ЖЦ ПК в **CMMI** сфокусировано на поэтапном определении, реально достигаемых результатов и на оценивании качества их выполнения. Качество процессов зависит от технологической среды, в которой они выполняются. ***Зрелость процессов*** – это степень их управляемости, возможность поэтапной количественной оценки качества, контролируемости и эффективности результатов – рис.1.3.

Два варианта модели CMMI – созданы для обеспечения ***непрерывного*** оценивания комплекса процессов в определенной области создания программных средств или для ***поэтапного*** оценивания и совершенствования зрелости предприятия, а также для организации ЖЦ комплексов программ в целом.


Уровень 5 – Оптимизационный – непрерывное улучшение:
управление изменением и качеством процессов и продукта;
инновации, количественное управление процессами и обеспечением ресурсами;
совершенствование качества и управления процессами и продуктами;
непрерывное измерение и улучшение качества продуктов и процессов.
Уровень 4 – Предсказуемый – количественное управление:
измерение характеристик компонентов, продукта и процессов;
количественное управление процессами, затратами и характеристиками качества;
распределение, контроль и обеспечение процессов ресурсами;
количественный контроль и управление рисками процессов, ресурсов и качества;
контроль и управление поставкой продуктов.
Уровень 3 – Определенный – стандартизация процессов:
определение стандартного процесса;
интегрированное управление проектом;
компонентами и продуктами;
верификация, валидация, контроль качества процессов и продуктов;
управление ресурсами, рисками и контроль их; документирование процессов и продуктов; обучение.
Уровень 2 – Управляемый - базовое управление:
управление процессами – управление требованиями; планирование и мониторинг проекта; управление разработкой компонентов и конфигурацией, обеспечение качества, контроль процессов и продуктов.

Рис. 1.3.

Модели CMMI представляют помощь специалистам при организации технологии и совершенствовании их продуктов, а также для упорядочения и обслуживания процессов разработки и сопровождения ПК. Концепция этих моделей покрывает управление и оценивание зрелости сложных систем, инженерии программных средств, а также процессов создания интегрированных программных продуктов и совершенствования их разработки. Компоненты непрерывной и поэтапной моделей в значительной степени подобны, могут выбираться и применяться в разном составе и последовательности использования в зависимости от свойств и характеристик конкретных проектов [29, 47].

Варианты описания моделей построены по единой схеме, которая содержит ***общие разделы***: предисловие; 1. введение; 2. модель компонентов; 3. терминология; 4. содержание уровней и главные компоненты каждого варианта модели (разработка целей и процедур); 5 раздел – структура взаимодействия процессов. Аннотированы четыре категории процессов раздела 7, их общий обзор и схемы взаимодействия CMMI процессов:

- менеджмент процессов;
- менеджмент – управление проектом;
- инжиниринг (технология);
- поддержка.

6 раздел – использование модели CMMI – краткие рекомендации для пользователей по применению модели и обучению; отмечается совместимость и соответствие процессов модели. Последний, ***седьмой раздел*** самый большой в каждом стандарте, он занимает около 500 страниц из полного объема документа, который составляет свыше 700 страниц.

В этом разделе представлены ***подробные рекомендации*** для реализации каждого из перечисленных в нем множества процессов, которые учитывают особенности конкретной модели.

Первый вариант (непрерывной) модели (***седьмой раздел) составляют процессы: менеджмент процессов; управление проектом; инженерия*** (технология); ***поддержка.*** В этой модели внимание акцентировано на организационных процессах, на планировании, управлении и контроле процессов реализации проектов программных комплексов, на разработке и управлении требованиями к программным продуктам.

Планирование проекта в первой модели, так же, как и во второй модели, включает:

- оценку возможного размера – масштаба программного продукта;
- оценку сложности функций и характеристик проекта ПК;
- определение модели и этапов жизненного цикла комплекса программ;
- технико-экономическое обоснование проекта – определение стоимости, трудоемкости и длительности ЖЦ ПК;
- разработка поэтапного графика работ и бюджета проекта;
- анализ, идентификация и оценка проектных рисков;
- планирование и управление документированием процессов и продуктов в ЖЦ проекта ПК;
- планирование и распределение технических и людских ресурсов по этапам ЖЦ ПС;
- планирование обеспечения знаний и квалификации коллектива специалистов для реализации проекта;
- обобщение и анализ совокупности планов проекта ПК;
- согласование работ и ресурсов по этапам ЖЦ разработчиком с заказчиком проекта ПК;
- документирование плана работ и утверждение его менеджером разработчиков проекта.

Процессы разработки требований к программному продукту аналогичны процессам в обеих моделях и включают:

- выявление реальных потребностей заказчика и пользователей к функциям и характеристикам программного продукта;
- разработку и согласование между заказчиком и разработчиком исходных, базовых требований к функциям программного продукта;
- определение доступных ресурсов и ограничений проекта комплекса программ;
- декомпозицию базовых исходных требований к функциям ПС в набор требований к компонентам и тестам комплекса программ;
- формализацию требований к интерфейсам между компонентами, с операционной и внешней средой;
- разработку концепции программного продукта и сценариев его использования;

• разработку требований к обобщенным характеристикам функциональной пригодности и использованию функций программного продукта по назначению.

Управление требованиями в обеих моделях включает:

• достижение однозначного понимания требований к проекту ПК заказчиком и разработчиками;

• получение заказчиком от разработчиков обязательств выполнить все его требования к программному продукту;

• согласованное между заказчиком и разработчиком управление изменениями требований к проекту ПК;

• обеспечение прослеживания корректности изменений от общих требований к проекту ПК до требований к компонентам и частным процессам;

• выявление и идентификация несоответствий между процессами разработки проекта и требованиями заказчика.

Второй вариант модели CMMI – ***поэтапное представление***. Первый уровень отличается значительной неопределенностью состава и содержания процессов в различных относительно простых проектах, поэтому он в документе не описан и не комментируется. Рекомендуется ограничиваться ***четырьмя (2-й – 5-й) основными уровнями*** – (см. рис. 1.3):

- ***второй уровень*** – формализует базовое управление проектами:
  - управление требованиями;
  - планирование проекта;
  - мониторинг и контроль проекта;
  - управление соглашениями с поставщиками;
  - измерение и анализ процессов и продуктов;
  - обеспечение качества процессов и продуктов;
  - управление конфигурацией;
- ***третий уровень*** – содержит стандартизацию основных процессов:
  - разработка требований;
  - технические решения;
  - интеграция продукта;
  - верификация;
  - валидация (аттестация);
  - определение организационных процессов;
  - интегрированное управление процессами и продуктами проекта;

- управление рисками;
- интегрированное управление поставщиками;
- анализ и разрешение проблем (устранение дефектов);
- организация окружения для интеграции;

- ***четвертый уровень*** – определяет количественное управление:
  - организация представления качества процессов;
  - количественное управление всем проектом и ресурсами;

- ***пятый уровень*** – оптимизационный, непрерывное совершенствование:
  - организация, инновации, количественное управление процессами и обеспечением ресурсами;
  - анализ причин дефектов, совершенствование качества и управления процессами и продуктами.

Рекомендуется на каждом более высоком уровне зрелости применять ***все процессы предыдущих нижних уровней***. В обоих вариантах модели каждый, выделенный выше базовый процесс, комментируется подробными рекомендациями для его практической реализации, которые содержат унифицированные по структуре описания:

- общие цели процесса, которые должны быть достигнуты;
- вводные замечания и общее описание функций процесса;
- специфические цели процесса;
- менеджмент процесса;
- разработка требований к процессу;
- взаимодействие и интерфейсы с другими процессами;
- практические цели – требуемые результаты действий процесса;
- планирование действий в определенном процессе;
- анализ и валидация (утверждение) результатов реализации процесса;
- мониторинг и контроль выполнения процесса.

Особое внимание в моделях CMMI уделяется процессам менеджмента проекта ПК. Эти ***требования и процессы моделей практически соответствуют***, регламентированным и детализированным рекомендациям в стандартах **ISO 9001:2000, ISO**. Требованиям к процессам в функциональных разделах 4 – 8 стандартов **ISO 9001, ISO 9004, ISO 90003** может быть сопоставлен адекватный по содержанию ряд разделов в ***моделях*** CMMI.

Для определения, представленных выше уровней зрелости процессов обеспечения жизненного цикла КП, разработан обширный

технический отчет **ISO 15504:1-5:2003-2006** – ***Оценка и аттестация зрелости процессов создания и сопровождения ПК и систем в составе пяти часте***й. Стандарт регламентирует оценку и аттестацию зрелости процессов создания, сопровождения и совершенствования программных средств и систем, выполняемых предприятиями. Применение стандарта направлено на выработку предприятиями и специалистами ***культуры постоянного совершенствования зрелости технологий*** обеспечения ЖЦ ПК, отвечающих бизнес-целям проектов и оптимизации использования доступных ресурсов.

Стандарты менеджмента (административного управления) качеством систем

Серия стандартов **ISO 9000:2000** разработана, чтобы помочь предприятиям всех типов и размеров внедрить и использовать эффективные ***системы менеджмента (административного управления) качества***. Совместно они образуют комплект согласованных стандартов управления системами качества и могут применяться как ***основа процессов управления в программной инженерии***:

- **ISO 9000:2000** – представляет введение в системы управления качеством продукции и услуг и словарь качества;
- **ISO 9001:2000** – устанавливает детальные требования для систем управления качеством, достаточные в случае необходимости продемонстрировать способность предприятия, обеспечить соответствие качества продукции и услуг требованиям заказчика;
- **ISO 9004:2000** – содержит руководство по внедрению и применению широко развитой системы управления качеством, чтобы достичь постоянного улучшения деловой деятельности и результатов предприятия.

Стандарты серии **ISO 9000:200**0 применяют процессный подход в административном управлении системами качества предприятий, а также рассматривают способы быстрого выявления и реализации возможностей для их улучшения. Процессная модель подчеркивает тот факт, что заказчики и другие заинтересованные стороны играют значительную роль в процессе установления исходных требований. После этого по отношению ко всем процессам, необходимым для создания необходимой продукции, применяется управление процессами и проводится проверка «выходов». Измерение степени удовлетворенности заказчика и других заинтересованных сторон используется в качестве обратной связи для оценки и признания того, что тре-

бования заказчика были выполнены полностью. Структура основных требований и рекомендаций в этих стандартах сведена к четырем объединенным крупным процессам:

- ***обязанности и ответственность администрации управления качеством*** (раздел 5);
- ***административное управление ресурсами*** (раздел 6);
- ***процессы жизненного цикла продукции и управления ее качеством*** (раздел 7);
- ***измерения, анализ и совершенствование продукции*** (раздел 8).

Стандарт **ISO 9001:2000** – Система менеджмента (административного управления) качества. Требования – является базой для Руководства по их реализации в стандарте **ISO 9004:2000** – ***Общие требования к системе менеджмента качества***. Организация – разработчик должна установить и управлять процессами, необходимыми для обеспечения уверенности в том, что продукция и/или услуга соответствуют требованиям заказчика. В качестве способа внедрения и демонстрации, установленных процессов, организация должна создать систему менеджмента качества, основываясь на требованиях настоящего международного стандарта.

Высшее руководство предприятия – разработчика **должно продемонстрировать свои обязательства заказчику относительно:**

- создания и поддержания важности удовлетворения требований заказчика;
- разработки политики качества и целей в области качества, а также планирования качества;
- создания системы менеджмента качества;
- проведения анализа деятельности со стороны руководства;
- обеспечения уверенности в наличии ресурсов.

Высшее руководство должно разработать политику в области качества ***и обеспечить уверенность в том, что она:***

- соответствует потребностям организации и её заказчиков;
- включает обязательства по удовлетворению требований и постоянному улучшению;
- обеспечивает основу для разработки и анализа целей в области качества;
- распространена, понята и внедрена во всей организации;

- анализируется с целью постоянного поддержания ее пригодности.

Система менеджмента качества – организация должна создать систему менеджмента качества, как средство реализации её политики в области качества, достижения своих целей в области качества и обеспечения уверенности в том, что продукция и/или услуга отвечает требованиям заказчика. Роли сотрудников и их взаимосвязи, а также ответственность и полномочия персонала должны быть установлены для того, чтобы способствовать эффективному управлению качеством, и должны быть доведены до соответствующих уровней организации. Высшее руководство должно уполномочить одного (или нескольких) лиц для:

- обеспечения уверенности в том, что система менеджмента качества внедрена и поддерживается в рабочем состоянии в соответствии с требованиями настоящего международного стандарта;
- доклада высшему руководству о функционировании системы менеджмента качества, включая вопросы, связанные с необходимостью ее улучшения;
- обеспечения уверенности в осознании во всей организации требований заказчика.

Организация должна разработать ***Руководство по качеству***, которое должно включать: описание элементов системы менеджмента качества и их взаимосвязей; общесистемные процедуры или соответствующие ссылки на них. Следует установить ***общесистемные процедуры для управления документами***, необходимыми для функционирования системы менеджмента качества.

Управление ресурсами необходимо для создания и поддержания в рабочем состоянии системы менеджмента качества. Организация должна проводить анализ и назначение персонала с целью обеспечения уверенности в том, что те, кто имеет обязанности, определенные системой менеджмента качества, являются компетентными для осуществления своей деятельности на основе соответствующего образования, подготовки, мастерства и опыта, создать и поддерживать в рабочем состоянии общесистемные процедуры по:

- определению потребностей в компетентном персонале и в его подготовке;
- обеспечению подготовки персонала в соответствии с выявленными потребностями;

- оценке через установленный период времени эффективности подготовки кадров;
- ведению соответствующих отчетов об образовании кадров, их подготовке, уровне мастерства и опыта.

Организация должна создать ***процесс идентификации требований заказчика*:**

- полноту требований заказчика к продукции;
- требования, не установленные заказчиком, но необходимые для применения продукции;
- обязательства по отношению к продукции, включая регламентирующие и законодательные требования;
- требования заказчика относительно пригодности продукции для её поставок и сопровождения.

Входные данные для проектирования и разработки ***должны включать*:**

- эксплуатационные требования заказчика или рынка;
- применяемые нормативные и законодательные требования;
- применяемые требования по охране окружающей среды;
- любые другие требования, существенные для проектирования и разработки.

Выходные данные процесса проектирования должны быть зарегистрированы в форме, дающей возможность проверки их по отношению к входным требованиям:

- соответствовать входным требованиям для проектирования и/или разработки;
- содержать или давать ссылку на критерий приемки продукции и/или услуги;
- определять характеристики продукции и/или услуги, которые являются существенными с точки зрения безопасности и правильного использования.

Утверждение проекта должно проводиться с целью подтверждения того, что конечная продукция способны отвечать требованиям для конкретных условий использования заказчиком. Когда это возможно, утверждение должно быть спланировано и выполнено до начала поставки или применения продукции. Результаты утверждения и последующих действий должны быть зарегистрированы.

Изменения проекта или модификация должны быть утверждены уполномоченным персоналом и зарегистрированы до их внедрения, следует определить влияние изменений на:

- взаимодействие между элементами проекта и/или разработки:
- взаимодействие между составными частями конечной продукции;
- имеющуюся продукцию и на функционирование ранее поставленной продукции;
- необходимость проведения повторной проверки или утверждения для всех или части выходных данных проектирования и/или разработки.

Организация должна ***спланировать и управлять производственными и сервисными операциями обслуживания***, включая те, которые предпринимаются после первоначальной поставки, посредством:

- предоставления технических условий, определяющих характеристики продукции, которые должны быть достигнуты;
- предоставления четко понимаемых производственных требований или инструкций для тех видов деятельности, где они необходимы для достижения соответствия продукции;
- внедрения надлежащих действий по мониторингу или проверке продукции;
- подходящих методов для выпуска, поставки и/или монтажа продукции.

Стандарт **ISO 90003:2004** – Рекомендации по применению стандарта **ISO 9001:2000** ***для программных продуктов*** – предназначены для регламентирования менеджмента при приобретении, поставке, разработке, применении, сопровождении ***сложных программных продуктов*** и при их обслуживании. Стандарт предлагается при установлении соответствия требованиям комплексов программ:

- как части коммерческого контракта с другими организациями;
- при представлении полезного продукта для рынка;
- для использования при поддержке процессов организации проектов ПК;
- для учета при встраивании программных продуктов в комплексы аппаратуры;
- при организации сервиса программных продуктов.

Полное или частичное применение стандарта **ISO 90003** целесообразно в различных ситуациях, с учетом технологии, модели жизненного цикла, процессов разработки, последовательности действий и организационной структуры предприятия. Структура стандарта **ISO 90003** привязана к разделам и требованиям в **ISO 9001:2000**, которые цитируются в начале каждого раздела. ***Пятый раздел*** определяет ответственность руководства: общие обязанности руководства управления проектом; политику в области обеспечения качества продукции; планирование управления проектом; ответственность и полномочия специалистов; анализ проектирования со стороны руководства.

В ***шестом разделе*** – менеджмент ресурсов, более полно комментируются особенности ***управления ресурсами в области программной инженерии***. Внимание акцентируется: на проблемах обеспечения ограниченными вычислительными ресурсами инфраструктуры проектов; на компетентности, квалификации и подготовке специалистов; на управлении производственной средой.

Наиболее обширным и специфическим, практически полностью ориентированным на программные продукты, является ***седьмой раздел*** стандарта. В нем изложено планирование и управление процессами и качеством жизненного цикла программных средств с попутными ссылками на рекомендации перечисленных выше стандартов.

Стандарт **ISO 19759:2005 – SWEBOK,** Совокупность знаний о разработке программных средств. ***Руководство*** – представляет фундаментальное, энциклопедическое обобщение современных процессов, методов и средств практического создания и обеспечения жизненного цикла сложных комплексов программ. В предисловии и введении изложены история развития, цели и функции ***практической программной инженерии***. Весь материал можно разделить на три части: базовые методы (разделы 1 – 6); процессы, технология и средства (разделы 7 – 12); приложения и комментарии (разделы A, B, C, D):

1. Введение в Руководство;
2. Требования к программным средствам;
3. Проектирование программных средств;
4. Конструирование программных средств;
5. Тестирование программных средств;
6. Сопровождение программных средств;
7. Управление конфигурацией программных средств;
8. Управление технологией создания программных средств;

9. Технологические процессы программных средств;
10. Технологические методы и средства;
11. Качество программных средств;
12. Взаимодействие дисциплин и технологий программных средств.

Внимание акцентируется на обеспечении эффективности процессов практической разработки и сопровождения комплексов программ. Важнейшими компонентами улучшения этих процессов являются ***количественные и качественные измерения результатов деятельности специалистов***, обучение и повышение их квалификации. Для этого необходимо управление персоналом, освоение новых технологий и инструментальных средств, контроль состояния, качества и изменения компонентов и комплексов программ, эффективное использование всех видов ресурсов. Каждый раздел основного текста содержит иллюстрации в виде схем и таблиц, список литературы и поддерживающих международных стандартов. В целом Руководство целесообразно использовать как учебное пособие для повышения системной квалификации специалистов по программной инженерии.

Глава 1.2

ПОДГОТОВКА КОЛЛЕКТИВА СПЕЦИАЛИСТОВ ДЛЯ ПРОЕКТИРОВАНИЯ И ПРОИЗВОДСТВА ЗАКАЗНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Основные свойства руководителей и специалистов, необходимые при проектировании и производстве заказных программных продуктов

Руководители и специалисты по проектированию заказных программных продуктов ***должны хорошо знать системотехнику производства больших компьютерных систем***, поскольку в них программный продукт играет определяющую роль. Технологии программной инженерии часто являются критическим фактором при разработке таких систем. Высокие темпы роста основных доступных ресурсов аппаратных средств, и сохраняющаяся потребность в увеличении использования их функций со стороны различных пользователей и сфер применения, приводят к необходимости ***адекватного совершенствования квалификации специалистов и технологий создания программных продуктов***. Это определяется ***интеллектуальными возможностями человека по интенсивности принятия творческих решений***. Им соответствуют наличие предельных значений производительности труда и длительности разработки сложных комплексов программ. Для их оценки, необходимо изучение и экстраполяция прецедентов и экспериментальных данных реальных разработок с наилучшими экономическими характеристиками, с учетом возрастания профессиональной квалификации специалистов и уровня автоматизации проектирования и производства.

При ***организации эффективного производственного коллектива*** следует учитывать конкретных цели специалистов, участвующих в проекте, их ***психологическую совместимость***, способность к дружной ной коллективной работе, наличие опыта взаимодействия в составе определенного коллектива, а также другие объективные и

субъективные свойства участников проекта. При этом большое значение может иметь личная мотивация и психологические особенности поведения разных специалистов при комплексной работе над сложным проектом. Эти характеристики могут быть обобщены в качественный показатель влияния сложности взаимодействия специалистов в коллективе, которому могут быть сопоставлены оценки возможного изменения эффективности производства. Наилучшим обычно считается непрерывное корректное взаимодействие организованных специалистов с большим опытом работы в конкретном коллективе при согласованности их целей, планов и методов работы. В остальных случаях в той или иной степени (даже в 3 – 5 раз) может возрастать трудоемкость разработки, что нельзя не учитывать при экономическом обосновании крупных проектов.

Необходимо обучение ***профессиональных коллективов специалистов*** современной ***программистской культуре*** промышленного создания крупных высококачественных программных продуктов – умению формализовать требования и достигать конкретные значения характеристик функционирования и применения сложных комплексов программ, с учетом тех ресурсов, которые доступны для обеспечения и совершенствования качества. Для этого при обучении необходимо воспитывать у каждого специалиста ***ряд коллективных профессиональных свойств*** – рис. 1.4 [1, 8, 11].

Способность критически оценивать конкурирующие решения – одна из основных черт, присущих профессиональному специалисту. Поэтому учебный план и методы преподавания должны помочь учащимся приобрести необходимые знания, навыки анализа и методы эффективной оценки. Важной их чертой является стремление к критическому мышлению. Необходимо также обучить умению оценивать надежность различных источников информации.

Оценка и оспаривание получаемых знаний – специалисты должны научиться не принимать на веру любую информацию от преподавателей или из книг. Они должны также понимать ограниченность современных знаний по программной инженерии и направления, в которых эти знания должны развиваться.

Осознание собственных ограничений вследствие чего профессионалы зачастую прибегают к консультировании у других профессионалов, а также показать преимущества командной работы.

Подготовка коллектива специалистов для проектирования и производства заказных программных продуктов включает:

- основные свойства руководителей и специалистов, при проектировании и производстве заказных программных продуктов;
  - знание системотехники производства больших заказных компьютерных систем и программных продуктов;
  - обеспечение психологической совместимости, способность дружной коллективной работы различных специалистов;
  - способность критически оценивать конкурирующие решения;
- подготовку и реализацию требований заинтересованных лиц к программному продукту;
  - корректное определение назначения, основных, необходимых и достаточных функции и характеристик конкретного КП;
  - оценка масштаба проекта, сопутствующих требований заказчика и доступных реальных ресурсов;
  - определение приоритетов заказчиков, потребителей и заинтересованных лиц при формировании требований к КП;
  - требования, которым обязан полностью удовлетворять программный продукт;
- требования к профессиональной квалификации руководителей для эффективного управления проектированием программных продуктов;
- требования к профессиональной квалификации руководителей и специалистов, организующих проектирование заказных программных продуктов;
- подготовку коллектива специалистов для проектирования компонентов сложных заказных программных продуктов.

Рис. 1.4

Эффективная коммуникация учит эффективно обмениваться информацией различными способами: письменно, при проведении презентаций, демонстраций собственных и чужих программных разработок, а также во время различного рода обсуждений, должны развить навыки восприятия на слух, навыки сотрудничества и переговоров.

Этичное и профессиональное поведение – специалисты должны научиться контролировать соответствие своей деятельности нормам этики, конфиденциальности и безопасности.

Стремление учиться и развиваться – так как большая часть изучаемого изменится в течение будущей профессиональной деятельности специалиста, и лишь малая часть необходимой информации преподается в вузе.

Специалист программной инженерии – это ***член команды – коллектива***, поэтому должен обладать ***навыками общения*** и межличностных отношений, а также уметь планировать не только свою работу, но и координировать ее с работой других специалистов. Известно, что индивидуальная производительность программистов может отличаться в десятки раз. Производительность одного и того же программиста в разных коллективах может так же отличаться в десятки раз. Поэтому основные усилия руководителя, если он стремится получить наивысшую производительность труда коллектива или рабочей группы, должны быть направлены на изучение и улучшение взаимодействия специалистов, обеспечивать эффективность работы каждого участника коллектива и обеспечивать эффективность процессов взаимодействия специалистов в коллективах разного размера и назначения.

Для того чтобы сотрудник мог эффективно ***решать поставленную задачу***, необходимо: понимание целей работы; умение ее делать; иметь возможности ее сделать и желание ее сделать. Для этого ***руководитель должен уметь*** эффективно выполнять свои функции и решать задачи: обеспечить сотрудникам общее видение целей и стратегии их достижения; обучать, быть наставником и образцом для подражания; помогать, обеспечить сотрудника всем необходимым, убрать препятствия с его пути; обеспечить адекватную мотивацию участника коллектива на протяжении всего проекта.

У каждого участника коллектива должна быть личная цель, которую он сможет достичь, продвигая производство комплекса программ. По ходу проектирования и производства мотивы людей, как правило, изменяются, самооценка, скорее всего, неадекватно завышается. Наиболее эффективные производственные процессы складываются в самоуправляемых и самоорганизующихся рабочих командах, для которых характерны ясность общих ценностей и целей, самоконтроль, взаимопомощь, взаимозаменяемость, коллективная ответственность за результаты труда, всемерное развитие и использование индивидуального и группового потенциалов.

Руководителю разработки программного продукта недостаточно быть хорошим управленцем, он должен стать ***признанным лидером***. Для этого необходимо: признание коллективом профессиональной компетентности и превосходства руководителя и доверие коллектива к действиям и решениям руководителя; признание его исключительных человеческих качеств, убежденность в его честности,

порядочности, вера в его искренность и добросовестность. В зависимости от состава, квалификации и ситуации задач коллектива, у лидера могут быть различные ***стратегии руководства при назначении***: руководителем в новый коллектив; в известный слаженный дружественный коллектив; в коллектив настроенный критически; где каждый сам себе может быть руководителем. Во всех ситуациях лидер обязан последовательно организовать эффективно действующий производственный коллектив [1, 10, 21].

Часто случается, что рабочая группа вязнет на одном из этапов и ***не достигает высокой производительности***. На совещаниях бесконечные неконструктивные споры и дискуссии. Постоянно доминируют одни и те же лица, другие предпочитают отмалчиваться. Все стараются избегать конфликтов и поддерживать согласие. При возникновении трудных ситуаций, все ждут решения от руководителя, редкие противоречия разрешаются общим голосованием. Подобное избегание производственных конфликтов снижает здоровую интеллектуальную конкуренцию, ведет к шаблонности и застою.

Альтернатива – когда ***в эффективном коллективе*** целенаправленно стимулируется творческое инакомыслие и интеллектуальная конкуренция. Назначают рецензентов – критиков, которые должны найти слабые стороны в решении, предлагаемом их коллегой для общего обсуждения, или поручают двум специалистам решить одну и ту же критичную для проекта задачу, используя разные подходы или технологии, а затем сравнить и оценить полученные результаты. Ни одно предлагаемое участником команды решение не принимается на веру, активно анализируются возможные негативные последствия или упущенные возможности при принятии решения. Конфликты носят исключительно производственный характер, а все, что человек делает с удовольствием, он делает максимально эффективно. В результате ***достигается высокая эффективность*** производственного коллектива.

В процессе работы над программными комплексами специалисты должны быть способными ***эффективно решать поставленные перед ними задачи*** как индивидуально, так и в команде. В реальной жизни специалистам предстоит выполнять некоторое количество проектов в одиночку или малыми группами, однако многие задачи требуют работы в большом коллективе с другими людьми. Соответственно, специалисты должны овладеть максимально полной информацией о сущности работы коллектива и ролях в команде. Они должны понимать важность та-

ких задач, как дисциплинированный подход, необходимость придерживаться установленных сроков и оценка как индивидуальной, так и командной производительности. Следует разрешать противоречия в стоящих перед проектом целях, находя приемлемые компромиссы в рамках существующих ограничений (стоимость, время, знания, существующие системы и организации). Специалисты должны выполнять задания, содержащие противоречивые и даже изменяющиеся требования.

Необходимо научиться использовать множество различных подходов к инженерному проектированию и к решению специфических проблем в конкретных предметных областях. Проектировать решения в одной или более предметных областях приходится, используя подходы программной инженерии, ***сочетающие и балансирующие этические, общественные, юридические и экономические интересы различных заинтересованных сторон***. Следует понимать достоинства и недостатки различных доступных альтернатив и последствия выбора того или иного подхода в каждой конкретной ситуации. В предлагаемых проектных решениях должны адекватно учитываться этические, общественные, юридические, экономические факторы, а также вопросы безопасности производства.

Следует ***понимать способности к ведению переговоров***, результативно работать, осуществлять руководство и эффективно общаться с заинтересованными лицами в типичных ситуациях при разработке программных продуктов. Программные инженеры должны осознавать, что им необходимо создавать программные комплексы, прежде всего, ***являющиеся полезными для заказчиков и пользователей***. Это должно способствовать получению опыта и созданию необходимой среды для подготовки высококвалифицированных специалистов по программной инженерии.

Подготовка и реализация требований заинтересованных лиц к программному продукту

Системная эффективность – функциональная пригодность комплекса программ может быть описана количественно или качественно, в виде набора полезных свойств и характеристик программного продукта, их отличий от имеющихся у других комплексов программ, а также источников возможной эффективности. Она определяет назначение, основные функции и требования заказчика, какие задачи должны решаться для удовлетворения пользователей, а допол-

нительные, конструктивные характеристики качества – как, и при каких условиях, заданные функции могут выполняться с требуемым качеством. В результате может быть формализована ***цель использования*** и набор главных характеристик, ***требований заказчика и пользователей*** при заказе или приобретении программного продукта, а также предполагаемая его сфера назначения и применения. Полнота и точность представления этих характеристик является исходной для прослеживания реализации всех последующих, производных свойств и качества функциональной пригодности комплексов программ (см. рис. 1.4).

Данное требование связано с тем, ***какие*** функции и задачи должен решать программный продукт для удовлетворения потребностей пользователей, в то время как другие, конструктивные требования главным образом связаны с тем, как и при каких условиях, заданные функции могут выполняться с требуемым качеством. Функциональная пригодность – это набор и описания атрибутов, определяющих ***назначение, основные, необходимые и достаточные функции программного продукта***, заданные техническим заданием и спецификациями требований заказчика или потенциального пользователя. Номенклатура и значения всех остальных показателей качества непосредственно определяются требуемыми функциями программного комплекса и, в той или иной степени, влияют на выполнение этих функций. Поэтому выбор функциональной пригодности, подробное и корректное описание ее свойств, являются основными исходными данными для установления требуемых значений всех остальных стандартизированных показателей качества. Атрибутами этой характеристики могут быть функциональная полнота решения заданного комплекса задач, степень покрытия функциональных требований спецификациями и их стабильность при совершенствовании программ, число и полнота реализуемых требований заказчика.

Функции и основные характеристики крупных комплексов программ условно можно отразить многомерными ***пространствами свойств и значений требований***, контроля и обеспечения их реализации.

Подготовка коллектива специалистов для проектирования и производства заказных программных продуктов включает:

- основные свойства руководителей и специалистов, при проектировании и производстве заказных программных продуктов;
  - знание системотехники производства больших заказных компьютерных систем и программных продуктов;
  - обеспечение психологической совместимости, способность дружной коллективной работы различных специалистов;
  - способность критически оценивать конкурирующие решения;
- подготовку и реализацию требований заинтересованных лиц к программному продукту;
  - корректное определение назначения, основных, необходимых и достаточных функции и характеристик конкретного КП;
  - оценка масштаба проекта, сопутствующих требований заказчика и доступных реальных ресурсов;
  - определение приоритетов заказчиков, потребителей и заинтересованных лиц при формировании требований к КП;
  - требования, которым обязан полностью удовлетворять программный продукт;
- требования к профессиональной квалификации руководителей для эффективного управления проектированием программных продуктов;
- требования к профессиональной квалификации руководителей и специалистов, организующих проектирование заказных программных продуктов;
- подготовку коллектива специалистов для проектирования компонентов сложных заказных программных продуктов.

Рис. 1.4.

В жизненном цикле крупных программных комплексов можно выделить три пространства требований, функции и характеристики которых, используются и взаимодействуют в следующих целях и назначении:

- исходные, ***утвержденные требования заказчика*** к функциям и характеристикам программного продукта, согласованные с разработчиками в виде конкретных документов, в соответствии с которыми разработчики обязаны создать и обеспечить применение программного продукта пользователями или в составе системы;
- ***реально реализованные разработчиками*** функции и характеристики программного продукта, которые обычно не могут полно-

стью и абсолютно точно соответствовать исходным требованиям, достоверно ***не полно известны*** заказчику, разработчикам и пользователям, и не отражены документами;

- реальные функции и характеристики программного продукта, которые ***практически используются пользователями*** и/или системой в соответствии с эксплуатационной документацией, и могут не совпадать с исходными реализованными требованиями, вследствие превышения их значений или не полного их использования.

Функции и характеристики в перечисленных областях обычно не совпадают, но в некоторой степени пересекаются и в совокупности отражают возможные ситуации положения пространства требований, которые полезно иметь в виду при организации и планировании тестирования и применения. Функции и характеристики программного продукта, которые достигаются реально разработчиками, обычно достоверно не известны или не используются заказчиком и пользователями, или могут находиться за пределами заданных исходных требований. Наибольшее значение для реального обеспечения качества программного продукта имеет ***полнота соответствия*** исходных требований к функциям и характеристикам, пространствам выполненных проверок при тестировании, адекватными им по содержанию и сценариям тестами. В результате совокупности ***требований к тестам*** могут применяться ***разработчиками как эталоны и вторая адекватная форма описания требований к комплексу программ*** для сквозной верификации спецификаций требований к тестам сверху вниз, а также сами подвергаться верификации на корректность соответствия исходным требованиям к функциям и характеристикам компонентов программ разного уровня.

Функциональные требования – определяют поведение программного продукта, который ***должен быть создан разработчиками*** для предоставления возможности выполнения заказчиком и/или пользователями своих обязанностей в контексте пользовательских требований. Выбор требований к характеристикам при проектировании программных продуктов начинается с определения исходных данных. Для корректного выбора и установления требований к характеристикам качества, прежде всего, необходимо определить основные особенности программной системы. На основе этих данных должен формироваться общий ***набор требуемых характеристик, свойств, их мер и значений качества для определенных потребителей в жизненном цикле программного средства.***

Проекты, как правило, инициируются заказчиком ***с объемом функциональных возможностей, превышающим*** тот, который разработчик может реализовать, обеспечив приемлемое качество при заданных ресурсах и сроках. Тем не менее, необходимо ограничиваться, чтобы иметь возможность предоставить в срок ***достаточно целостный и качественный программный продукт***. Существуют различные методы задания очередности выполнения (приоритеты) требований и понятие базового уровня – совместно согласованного представления о том, в чем будут состоять ключевые функции системы, как продукта проекта – понятие состава требований, задающих ориентир для принятия решений и их оценки [10, 31].

Приоритеты потребителей при формировании требований к комплексу программ отражаются не только на выделении важнейших для них критериев и ранжировании приоритетов других характеристик, но также на возможности исключении из анализа некоторых характеристик качества, которые для данного потребителя не имеют значения. После определения назначения и функций подготовка исходных данных и концепции комплекса программ должны завершаться ***выделением номенклатуры приоритетных функций и характеристик качества***, имеющих достаточное влияние на функциональную пригодность ***для определенных потребителей***. Требования к значениям функций и характеристик программного продукта должны быть предварительно проверены разработчиками на их реализуемость с учетом доступных ресурсов конкретного проекта и при необходимости откорректированы по составу и значениям с учетом рисков. При ограниченности ресурсов проекта распределение приоритетов должно становиться более строгим и могут снижаться приоритеты характеристик, для реализации которых ресурсов недостаточно.

При определении критериев качества требований следует ***выявить и прояснить нечеткие требования***. Для них в одних случаях можно определить согласованный критерий качества, а в других случаях получается критерий, по которому не достигнуто ***соглашение с заказчиком***. Такое нечеткое требование целесообразно заменить несколькими требованиями, каждое из которых будет иметь свой собственный критерий качества. Не каждое требование может иметь критерий качества, который можно использовать для проверки того, удовлетворяет ли какое-либо решение требованию. Добавив критерий качества для каждого требования, иногда можно сделать их ося-

заемыми, понятными. Это первый шаг по определению критериев для измерения качества решений.

Когда формулируются требования помимо тех, которые уместны для назначения проекта, есть вероятность, прихватить и несущественные требования, которые часто появляются в результате ***непонимания заинтересованной стороной целей проекта***. Если заинтересованной стороне нужно что-то особенное, она полагает, что это является требованием, даже если это несущественно для проекта системы. Чтобы протестировать ***требования на значимость***, следует сопоставить требования и сформулированные цели разработки программного продукта:

- способствует ли требование достижению целей продукта;
- если исключить требование, помешает ли это достижению целей продукта;
- существуют ли другие требования, которые зависят от данного.

Необходимо уметь проверять, что разрабатываемый комплекс программ удовлетворяет каждому из зафиксированных и утвержденных требований. Нужно выделять каждое такое требование, чтобы ***проследить его развитие*** в ходе подробного анализа, проектирования и, наконец, реализации. Каждый этап разработки комплекса программ формирует, уточняет и реорганизует требования, чтобы сделать их как можно ближе к назначению нового программного продукта.

Каждое требование должно отражать ***распознаваемую, измеряемую сущность***. Необходимо определять и фиксировать связи между требованиями и влияние одних требований на другие. В проектах крупных комплексов программ нужно применять способ работы с большим числом требований и сложными связями между ними. Следует учитывать, что наряду с существованием некоторого числа требований, связанных с одним событием и/или сценарием использования, любое требование может быть связано с другими событиями и/или сценариями использования продукта. Событие и/или сценарий использования требований предоставляют некоторое количество небольших взаимосвязанных систем. С помощью конфигурационного управления требованиями можно прослеживать, с какими событиями и/или сценариями использования какие решения связаны. Если в требование вносится изменение, следует определить части системы, на которые влияет это изменение.

Спецификация комплекса программ должна содержать все ***требования, которым обязан удовлетворять программный продукт***. В спецификации необходимо объективно определить все, что он должен делать, а также те условия внешней среды, при которых он должен применяться и функционировать. Наиболее ответственный аспект формирования требований – ***общение со специалистами***, которые вносят требования. При наличии согласованного способа формирования требований все заинтересованные стороны могут принимать участие в процессе определения требований. Как только сформулировано хотя бы одно требование, можно приступать к его тестированию на корректность и задавать заинтересованным сторонам подробные ***вопросы для уточнения и конкретизации тестов***. Целесообразно заказчику определять относительную значимость, приоритеты требований, задавать критерий качества для каждого требования и использовать этот критерий для тестирования корректности возможных решений.

Требования к профессиональной квалификации руководителей и специалистов, проектирующих программные продукты

Стоимость аппаратных средств сложных технических систем постепенно снижается, тогда, как ***стоимость программных продуктов быстро возрастает***. Возрастают как размеры программных продуктов, так и их сложность. Это является одной из причин возникновения проблем при разработке крупных программных продуктов, как и то, что многие компании, занимающиеся их производством, не уделяют должного внимания эффективному ***применению специалистами современных методов и стандартов***, разработанных в программной инженерии. Менеджеры и коллективы специалистов должны понимать, что они работают в организационных и финансовых рамках заключенных контрактов, и должны решать поставленные задачи ***с учетом условий и требований договора – контракта на систему***. Программирование компонентов – это дело, главным образом, индивидуальное, а программная инженерия систем – всегда коллективная работа.

Менеджеры, руководители – разработчики должны применить методы и процессы для того, чтобы ***понять решаемую проблему заказчика*** до начала производства программного продукта (см. рис.1.4).

Для этого следует использовать ***анализ, выявление и освоение проблемы и интересов заказчика***:

- достигнуть соглашения между заказчиком и разработчиком по определению проблемы, целей и задач продукта;
- выделить основные причины – проблемы, их источники, стоящие за основной проблемой проекта системы и программного продукта;
- выявить заинтересованных лиц и пользователей, чье коллективное мнение и оценка в конечном итоге определяет успех или неудачу программного продукта;
- определить, где приблизительно находятся область и границы возможных решений проблем;
- понять ограничения, которые будут наложены на проект, коллектив специалистов при решении проблем.

Понимание потребностей пользователей необходимый организационный этап, так как разработчики редко получают совершенные спецификации требований к создаваемой системе, квалифицированные руководители должны сами ***добывать*** информацию, необходимую им для успешной работы. Термин ***выявление требований*** точно отражает данный процесс, в котором разработчики должны играть активную роль. Чтобы помочь коллективу специалистов решить эти проблемы, лучше понять потребности пользователей и других заинтересованных лиц, целесообразно использовать ***методы*** [1, 30]:

- интервьюирования и анкетирования – создание структурированных интервью;
- проведение интервью с 5 – 15 пользователями и/или заинтересованными лицами;
- подведение итогов совокупности интервью, формулирование 10 – 15 наиболее часто упоминавшихся функциональных и архитектурных потребностей заказчика и пользователей;
- совещания, посвященные анализу и синтезу требований – формулирование и определение целей программного продукта;
- ознакомление с ними всех участников проекта и установление, что они с ними согласны; если это не так, следует остановиться и уточнениями добиться согласия; обязательно убедиться в согласии заказчиков;
- анализ иллюстративных прецедентов в приложении к концепции требований, чтобы их функции были наглядны и понятны;

- по возможности выявление или создание временных прототипов на основе первичных требований.

Хотя ни один из методов не является универсальным, каждый из них позволяет лучше ***понять потребности пользователей*** и тем самым превратить неясные требования, в требования, которые известны и понятны. Каждый из этих методов эффективен в определенных ситуациях, однако целесообразно отдавать предпочтение совещаниям, посвященным требованиям[1, 21, 35].

При первоначальном определении требований к функциональной пригодности и к конструктивным характеристикам качества, ***заданные заказчиком ограничения ресурсов*** не всегда могут учитывать ряд особенностей проекта, что обусловит недопустимое снижение (или завышение) требований к некоторым характеристикам качества. Кроме того, возможно, что некоторые характеристики противоречивы или принципиально нереализуемы в данном проекте. В результате ***не сбалансированные требования*** к качеству и доступные ресурсы проявятся как риски – ущерб в виде потерь в качестве или в излишнем расходовании ресурсов. Для устранения или снижения рисков до допустимых пределов потребуется изменение требований к функциональной пригодности и/или к конструктивным характеристикам.

Системные, функциональные требования – должны отражать высокоуровневые требования заказчика к задачам и функциям крупного программного комплекса, содержащего ряд взаимосвязанных подсистем. При этом, система может быть как целиком программной, так и состоять из программной и аппаратной частей. В общем случае, частью системы ***может быть персонал***, выполняющий определенные функции системы.

Программные средства все больше встраиваются в различные системы. Работа с такими проектами требует от менеджера и программного инженера широкого взгляда на общие ***задачи проектирования систем***. Им необходимо участвовать в выработке требований для всей системы, а также освоить прикладную область, требованиям которой должен будет отвечать программный продукт.

Программные ***руководители и системные аналитики*** должны быть знаком с основными способами проектирования сложных систем, знать, ***как перевести расплывчатые требования и пожелания заказчика системы в четкое техническое задание***, и уметь разговаривать с заказчиком и потенциальными пользователями системы на языке предметной области, а не на профессиональном программист-

ском жаргоне. Такие способности требуют, в свою очередь, гибкости и открытости, чтобы выделять сущность предметной области программных комплексов и систем.

Для конкретного комплекса программ доминирующие требования выделяются и определяются его ***функциональным назначением***. Программные продукты для ЭВМ как объекты проектирования, производства, испытаний и оценки качества характеризуются в жизненном цикле следующими ***обобщенными характеристиками***:

- проблемно – ориентированной областью применения, техническим и социальным назначением программного продукта;
- конкретным классом и назначением решаемых функциональных задач с достаточно определенной областью применения соответствующими пользователями;
- масштабом и сложностью совокупности программ и базы данных, решающих единую целевую задачу системы;
- архитектурой комплекса программ и базы данных;
- необходимыми составом и требуемыми значениями характеристик качества функционирования программ и величиной допустимого ущерба – риска из-за недостаточного их качества;
- составом потребителей характеристик комплекса программ, для которых важны соответствующие атрибуты качества;
- комплектом стандартов и их содержания, которые целесообразно использовать при выборе характеристик комплекса программ;
- реальными ограничениями всех видов ресурсов и сроков проекта;
- степенью связи решаемых задач с реальным масштабом времени или допустимой длительностью ожидания результатов решения задач;
- прогнозируемыми значениями длительности эксплуатации и перспективой создания, множества версий программного продукта;
- предполагаемым тиражом производства и применения комплекса программ;
- степенью необходимой документированности программного продукта.

Исходные требования к комплексу программ могут быть представлены и согласованы в ***составе спецификации всей системы***. Если программный продукт нуждается во взаимодействии с другими программными или аппаратными продуктами, то в спецификации требований должны быть оговорены непосредственно или при помо-

щи ссылок интерфейсы между разрабатываемыми и другими применяемыми продуктами. При ***заключении контракта спецификация сложных требований*** может быть определена не полностью, она может быть доработана в ходе реализации проекта. При разработке заказчиком и руководителем проекта требований технического задания на крупный комплекс программ должны учитываться ***общесистемные ограничения, которыми могут быть*** [16, 24, 42]:

- экономические – финансовые или бюджетные ограничения ресурсов; себестоимость и ценообразование; проблемы лицензирования;
- политические – внешние или внутренние политические ограничения предприятия, влияющие на потенциальное решение; проблемы в отношениях между подразделениями;
- технические ограничения проекта в выборе технологий; требования работать в рамках существующих платформ или технологий; запрет использования любых новых технологий; ограничения использовать закупаемые пакеты программной индустрии;
- системные требования обеспечивать совместимость с существующими решениями и системами; применение определенных операционных систем и внешней среды комплекса программ;
- эксплуатационные ограничения информационной среды; правовые или юридические ограничения; требования безопасности; ограничения требованиями стандартов;
- ресурсы – ограничения сроками и графиком проекта; ограничения доступными техническими и кадровыми ресурсами; отсутствие возможности привлечения дополнительных специалистов.

При создании высококачественных комплексов программ, прежде всего, ***необходима организация и тесное взаимодействие представителей заказчика и руководителей*** проекта. Взгляды и требования заказчика, в основном, отражаются в функциональных и потребительских характеристиках версий программного продукта. Устремления разработчиков направлены на возможность и способы их реализации с требуемым качеством. Эти различия исходных точек зрения на проект приводят к тому, что некоторые неформализованные представления тех и других имеют ***зоны неоднозначности и взаимного непонимания*** требований к продукту, что может приводить к конфликтам при выборе квалифицированного коллектива специалистов. Организация четкого взаимодействия и сокращение этих зон неоднозначностей требует проведения определенных мероприятий и

контактов по обмену знаниями, взаимному повышению квалификации и обучению. Представители заказчика, участвующие в прогнозировании проекта, должны обучаться формализации автоматизируемых технологических процессов производства, для которых предназначены соответствующие инструментальные средства, и иметь представление об эффективных путях их применения.

Для планирования и управления жизненным циклом концептуально целостных программных продуктов и обеспечения их качества, необходимы организационные действия системных архитекторов, направленные на ***подбор и обучение коллектива специалистов разных категорий и специализаций***. Руководство производством крупных программных продуктов целесообразно осуществлять двум лидерам – менеджерам с различными функциями:

- ***менеджер – руководитель проекта*** – этот специалист, обеспечивающий коммуникацию между заказчиком и коллективом специалистов, его задача – определять и обеспечивать удовлетворение требований заказчика с учетом доступных ресурсов;
- ***менеджер-архитектор программного продукта*** – должен управлять коммуникациями и взаимоотношениями в большом производственном коллективе, являться координатором последовательности создания компонентов, разрабатывать базовые, функциональные спецификации требований и управлять ими, вести график проекта и отчитываться за его состояние и развитие, инициировать принятие критичных решений.

Руководителем проектирования программного продукта в зависимости от особенностей продукта может быть: менеджер продукта, менеджер проектирования, руководитель проекта. ***Лидер – руководитель проектирования должен уметь*** [16, 26, 30]:

- руководить процессом выявления и формирования требований к функциям программного продукта, подлежащим производству;
- вести переговоры с руководством системы, пользователями и разработчиками компонентов и поддерживать равновесие между тем, чего хочет заказчик, и тем, что может создать команда разработчиков за ресурсы и время, отведенные для их реализации;
- рассматривать конфликтующие пожелания, поступающие от различных участников проекта комплекса программ и находить компромиссы, необходимые для определения приоритетов набора функций, представляющих наибольшую ценность для максимального числа участников проекта, прежде всего, заказчика и пользователей;

- осуществлять проверку спецификаций требований программного комплекса, чтобы удостовериться, что они соответствуют базовой концепции проекта и функций программного продукта;
- осуществлять управление изменением приоритетов задач и функций, а также добавлением и исключением новых функций комплекса программ.

Управление специалистами при проектировании и производстве включает хранение версий требований к компонентам, отслеживание связей, оценку рисков требований, выстраивание последовательности приоритетов требований. Группа поддержки руководителя должна уметь описать способ управления требованиями к продукту и их реализацией ***в плане производства***. Ключевые соображения при выборе способа хранения требований должны быть – простота и гибкость сортировки требований и подготовки отчетов, простота и скорость доступа к требованиям, эффективность сопровождения и модификации требований. Группа управления должна составлять график разработки и выполнения процедур, чтобы определить время и ресурсы на эти работы. ***При подготовке руководителем проектирования и производства должны***:

- назначаться сотрудники, ответственные за выполнение каждой из работ и компонентов;
- учитываться последовательность разработки компонентов, их взаимозависимость и связь с процессами и другими компонентами комплекса программ;
- сокращаться или исключаться возможные конфликты между процедурами производства, которые должны быть документированы;
- процедуры производства должны быть объединены в группы в соответствии с функциями программного продукта;
- разработка производственных процедур и их применение должны быть спланированы таким образом, чтобы исключить дублирование работ;
- в структуре процедур следует учитывать и документировать их приоритеты и риски;

Руководителю проектирования комплекса программ необходимо уметь описать ***последовательность*** выполнения процедур производства и их ***взаимозависимость***, поскольку определенная функция зачастую не может быть выполнена, пока предыдущая функция не сформирует нужные данные. Необходимо планировать работы по на-

стройке среды, подготовке обязательных отчетов в график разработки и выполнения процедур. График разработки и выполнения производственных процедур должен позволять так организовать действия, чтобы один специалист не мог независимо изменять данные, используемые другим специалистом.

Состав основных требований к программному продукту:

- описание обобщенных результатов обследования и изучения существующей системы и внешней среды;
- описание целей, назначения программного продукта и потребностей заказчика и потенциальных пользователей к нему в заданной среде применения;
- перечень базовых стандартов для предполагаемого проекта программного продукта;
- общие требования к характеристикам комплекса задач программного продукта:
  - цели создания программного продукта и назначение комплекса функциональных задач;
  - перечень объектов среды применения (технологических объектов управления, подразделений предприятия и т. п.), при управлении которыми должен решаться комплекс задач;
  - периодичность и продолжительность решения комплекса задач;
  - связи и взаимодействие комплекса задач с внешней средой и другими компонентами системы;
  - распределение функций между персоналом, программными и техническими средствами при различных ситуациях решения требуемого комплекса функциональных задач;
- требования к входной информации:
  - источники информации и их идентификаторы;
  - перечень и описание входных сообщений (идентификаторы, формы представления, регламент, сроки и частота поступления);
  - перечень и описание структурных единиц информации входных сообщений или ссылка на документы, содержащие эти данные;
- требования к выходной информации:
  - потребители и назначение выходной информации; перечень и описание выходных сообщений; регламент и периодичность их выдачи;

- допустимое время задержки решения определенных задач;

- сопоставительный анализ требований заказчика и возможностей пользователей к программному продукту набора функций для удовлетворения требований заказчика и пользователей;
- обоснование выбора оптимального варианта требований к содержанию и приоритетам функций комплекса;
- общие требования к структуре, составу компонентов и интерфейсам с внешней средой;
- ожидаемые результаты и возможная эффективность реализации выбранного варианта требований;
- общие требования к составу и содержанию документации программного продукта;
- оценка необходимых затрат ресурсов на разработку, ввод в действие и обеспечение функционирования программного продукта;
- предварительный состав требований, гарантирующих качество применения программного продукта;
- предварительные требования к условиям испытаний и приемки системы и программного продукта.

Требования к профессиональной квалификации руководителей и специалистов, организующих проектирование заказных программных продуктов

Квалифицированные руководители проектирования и производства программных продуктов должны уметь осуществлять детальное планирование, распределение этапов и трудоемкость работ, времени для их выполнения, квалифицированными специалистами и других ресурсов. ***Выделенный для проектирования руководитель*** должен отвечать за планирование и управление проектом, работами и задачами реализации планов производственных процессов, таких как заказ, разработка, поставка, эксплуатация, сопровождение и ряд вспомогательных процессов. Он должен уметь определить экономические возможности реализации спланированных процессов, проверяя наличие, соответствие и достаточность ресурсов, выделенных для выполнения и управления ***каждым процессом*** – квалифицированных специалистов, технологии и условий внешней среды, а также реальность ***сроков завершения производства*** программного продукта.

Руководителю следует установить и поддерживать в рабочем состоянии документированные процедуры, гарантирующие производство компонентов и программного продукта в соответствии с задан-

ными требованиями и согласно плану. Такие планы должны регламентировать виды деятельности или содержать ссылки на стандарты и ***определять ответственных лиц*** за их осуществление. В планах необходимо определять, кому, как и каким образом, следует управлять производством конкретных компонентов продукта, анализировать выполнение работ, а также установить виды и частоту отчетов специалистов для руководства, потребителей и других заинтересованных сторон, принимая во внимание все требования заказчика. В планах должны определяться производственные подразделения и состав специалистов, которые будут исполнять конкретные виды работ.

Следует уметь определить этапы разработки и то, какие результаты и документы – на компоненты, подсистемы или версии программного продукта должны быть получены по окончании этих этапов. Далее начинается итерационная часть планирования. Сначала разрабатывается ***укрупненный план этапов работ*** по созданию продукта. После этого проводится оценка возможности выполнения сроков работ и отмечаются расхождения между требуемым и планируемым временем завершения работ. По мере поступления новой информации о ходе выполнения проекта возможен пересмотр первоначальных оценок плана и базовых требований к всему продукту. Это, в свою очередь, может привести к изменению функций, размера продукта и графика работ.

Представленные выше категории специалистов, участвующих в создании крупных программных продуктов, необходимо уметь использовать в составе больших скоординированных коллективов, объединенных решением ***единой целевой задачей***. Для этого должна быть определена и зафиксирована роль каждого специалиста в коллективе, что, когда и какого качества производственные процессы следует выполнить во взаимодействии с остальными специалистами проекта. ***Руководитель – менеджер проекта*** должен уметь выполнять координацию деятельности специалистов, ***для чего подготовить планы*** выполнения всеми специалистами производственных процессов, которые должны содержать описания соответствующих задач и работ, а также обозначения создаваемых программных компонентов и продуктов. В ***обобщенном плане руководители должны уметь определять*** [21, 37]:

- предварительные графики своевременного решения специалистами конкретных функциональных и вспомогательных задач;
- оценки трудозатрат, времени и ресурсов на их решение;

- распределение задач по специалистам, их квалификация и обязанности;
- используемые в производственных процессах критерии управления качеством реализации компонентов;
- условия и производственная инфраструктура выполнения специалистами запланированных процессов и функций.

Руководитель должен ***уметь подготовить укрупненные планы по каждому виду профессиональной деятельности квалифицированных специалистов*** и утвердить содержание крупных детализированных планов по компонентам и видам деятельности. Должно быть установлено организационно-техническое ***взаимодействие между различными группами специалистов***, которые вносят свой вклад в процессы и обеспечение качества производства, а необходимая информация должна документироваться и регулярно анализироваться. В планах работ поставщиков и субподрядчиков ***надо уметь*** определить ***границы ответственности*** за каждую часть программного продукта и за способ обмена технической информацией между всеми сторонами.

Для планирования производства сложных программных продуктов целесообразно ***уметь обобщать и использовать экспериментальные статистические распределения основных экономических характеристик*** – трудоемкости, длительности и числа специалистов по этапам работ и по реальному времени реализации компонентов ***предшествующих проектов***. В оценках совокупных затрат на производство полностью новых комплексов программ доминирует трудоемкость (стоимость) непосредственной разработки программных компонентов. Для этих распределений характерно ***наличие максимума*** в использовании трудовых ресурсов на средних этапах разработки – ***на этапах программирования и автономной отладки компонентов***. Эти этапы полностью заняты производством компонентов, а на остальных этапах к компонентам относится только часть таких затрат, а также интегрирования компонентов и испытаний комплексов программ.

Важно уметь на каждом этапе выделять долю труда, которую можно отнести к тестированию и отладке компонентов, их сборке, комплексированию и испытаниям крупных функциональных частей комплекса программ. ***На программирование и автономную отладку программных компонентов*** требуется обычно около ***половины*** трудоемкости и до 40% общей длительности разработки. Возрастание

относительных затрат всех видов при комплексной отладке в реальном времени обусловлено высокой трудоемкостью динамического тестирования и испытаний для этого класса программных продуктов. Относительная трудоемкость и длительность при испытаниях различных классов программных продуктов приблизительно одинаковая и составляет 8 – 10% от совокупных затрат на производство. Однако абсолютная трудоемкость для испытаний программ, не связанных с реальным временем, приблизительно в 5 раз меньше, чем программ, требующих динамических испытаний, а абсолютная длительность этого процесса сокращается всего в полтора раза.

Совокупные относительные затраты ***на автономную и комплексную отладку, а также на испытания*** крупных динамических комплексов программ приблизительно одинаковые, и составляют ***по трудоемкости и по длительности около 30%***. Эти соотношения имеют тенденцию медленно снижаться за счет развития современных технологий, инструментальных средств автоматизации и повышения качества начальных этапов проектирования и производства. Необходимость проведения комплексирования и динамического тестирования в реальном времени может привести к меньшему улучшению экономических характеристик для класса программ реального времени при активном внедрении средств автоматизации разработки.

Руководитель должен ***уметь оценивать*** длительность этапов и всего проекта, определять виды и размеры ресурсов, необходимых для реализации отдельных этапов и типов работ, и представлять их в виде согласованной с заказчиком графиков последовательности проектирования и производства. Если данный проект подобен ранее реализованному, то график производства нового проекта можно взять за основу. Если проект является инновационным, первоначальные оценки длительности и требуемых ресурсов почти наверняка будут ***слишком оптимистичными***, даже если планировщик попытается предусмотреть все возможные неожиданности. С этой точки зрения проекты производства программных продуктов не отличаются от больших инновационных технических проектов, в графиках которых неожиданно могут возникать проблемы и трудности. Именно поэтому ***графики работ являются динамичными*** и их необходимо постоянно контролировать и обновлять по мере поступления новой информации о ходе выполнения проекта.

Экономическое обоснование проектов квалифицированными заказчиками и производителями на начальном этапе их проектиро-

вания должно содержать оценки рисков реализации поставленных целей, обеспечивать возможность планирования и выполнения жизненного цикла программного продукта или указывать на недопустимо высокий риск его реализации и целесообразность прекращения разработки. Большую часть рисков и негативных последствий производства можно избежать, освоив и используя существующие ***методы оценивания и прогнозирования производственных затрат,*** а также управления проектами программных продуктов для их успешного завершения.

Для заказчика и разработчиков при заключении контракта необходимо ***умение квалифицированно выполнять*** достаточно достоверное прогнозирование требований к программному продукту и ***экономическое обоснование необходимых ресурсов*** по трудоемкости, стоимости, срокам и другим показателям. Заказчик заинтересован в получении продукта высокого качества при минимальных затратах, а разработчик желает получить максимальную оплату за созданный продукт и достаточные ресурсы на его производство. Противоположность интересов поставщика и потребителя при оценке экономических характеристик, стоимости и других ресурсов проекта, требует поиска компромисса, при котором производитель программного продукта не продешевит, а заказчик не переплатит за конкретные выполненные работы и весь проект. Поэтому оба партнера ***заинтересованы в квалифицированном экономическом прогнозировании*** и обосновании проектирования и производства программного продукта.

Экономические характеристики производства – стоимость, трудоемкость, длительность комплексов программ, в ***основном определяет человеческий фактор*** – количество, квалификация и организация коллектива специалистов. Эти экономические характеристики обычно являются ***исходными требованиями***, выдаваемыми заказчиком как доступные, ограниченные ресурсы для разработчиков. Для обеспечения производства программного продукта в таких заданных условиях необходим коллектив специалистов, имеющих соответствующую профессиональную квалификацию. Это приводит к необходимости оценивания ***требований к коллективу специалистов***, необходимого для реализации определенного программного продукта, исходя ***из заданных заказчиком экономических характе-ристик***. Альтернативой обычно является оценка возможных экономических характеристик производства требуемого программ-ного продукта в

зависимости от человеческих факторов коллектива предприятия, инструментальной и внешней среды производства.

Большую роль в повышении экономической эффективности производства программных продуктов играет ***повторное использование готовых программных компонентов*** (ПИК) из других проектов. При этом значительно сокращаются этапы программирования и автономной отладки модулей и компонентов программ, а также в той или иной степени длительность других этапов. Поэтому зависимость длительности, от доли ПИК оказывается нелинейной, и заметное сокращение длительности разработки проявляется только при создании базовой версии программного продукта практически полностью из готовых компонентов.

Подготовка специалистов для проектирования компонентов сложных заказных программных продуктов

Анархичные подходы к написанию программных компонентов завоевали устойчивую репутацию ***основного источника ошибок, проблем и дефектов в сложных комплексах программ***. Результатом этого служат возрастающие ***требования заказчиков к стандартизированной организации процессов проектирования компонентов*** программного продукта у поставщиков, причем без удовлетворения этих требований зачастую невозможно получить контракты. В частности, после потерь миллионов долларов, выплаченных за неработающие или незаконченные программные продукты, государственные органы США начали требовать от всех своих подрядчиков соответствия, как минимум ***третьему уровню зрелости*** модели CMMI (см. главу 1.1). Для этого подрядчики должны наладить стабильный, надежный и устойчивый процесс проектирования и производства компонентов и программных продуктов. Специалисты должны понимать значимость налаживания личных контактов с заказчиками, согласования детальных требований к будущим программам и необходимых усилий по достижению наивысшего уровня качества компонентов и КП.

Поэтому специалисты, участвующие в ***проектировании сложных заказных проектов должны***:

- владеть знаниями и навыками программной инженерии, необходимыми для того, чтобы приступить к практическому проектированию;
- уметь работать индивидуально или в группе над созданием проектов качественных программных компонентов;

- искать приемлемые компромиссы в рамках ограничений, накладываемых затратами, временем, срокам, знаниями, существующими системами и организацией;
- уметь выполнять проектирование компонентов в одной или в нескольких предметных областях, используя подходы программной инженерии, объединяющие функциональные, этические, социальные, юридические и экономические интересы;
- демонстрировать понимание и применение существующих теорий, моделей и методов, современной программной инженерии;
- демонстрировать такие навыки, как межличностное общение, эффективные методы коллективной работы, лидерство;
- изучать новые модели, методы и технологии проектирования и производства комплексов программ по мере их появления.

Специалисты должны быть ориентированы на коллективную, групповую работу над компонентами программных комплексов. Внимание их должно акцентироваться ***на комплексе индустриальных методов и международных стандартов программной инженерии,*** которые непосредственно обеспечивают эффективный жизненный цикл сложных высококачественных программных продуктов. Четкая формулировка моделей компонентов существенно упростит взаимообмен стратегиями и инструментами между коллективами и отдельными специалистами. Следует определять фундаментальные навыки и знания, которыми необходимо обладать всем, специализирующимся на проектировании компонентов программной инженерии.

Быстрая эволюция программной инженерии, стандартов и особенности профессиональной деятельности требуют постоянного ***обновления квалификации специалистов***. При проектировании компонентов необходимо учитывать изменения в технологиях, методиках и приложениях, новые разработки в сфере проектирования, а также важность концепции ***«обучения специалистов на протяжении всей жизни»***. В такой быстро развивающейся области, как программная инженерия, предприятия должны оперативно перенимать передовые стратегии, реагируя на происходящие изменения. Кроме того, освоение программной инженерии должно готовить специалистов к дальнейшему обучению на протяжении всей жизни, что позволит им идти в ногу со временем и быть способными разрешать сложные проблемы будущего.

Возрастание сложности и ответственности современных задач, решаемых заказными компонентами программ реального времени, а также возможность большого ущерба от недостаточного качества их

результатов, значительно повысило актуальность освоения специалистами методов полного, ***стандартизированного описания требований*** к характеристикам качества проектирования компонентов программ на различных этапах жизненного цикла. Необходима систематизация реальных характеристик качества, а также обучения специалистов применению стандартов, выбора из них и адаптации необходимой номенклатуры и требуемых значений характеристик для конкретного проектирования компонентов и комплексов программ.

Подготовка и обучение специалистов методам и процессам ***проектирования и освоения требований*** к компонентам сложных комплексов программ включают:

- основы инженерии требований к программному продукту и его компонентам;
- процессы инженерии требований: выявление требований, спецификация, анализ и управление;
- анализ требований: инспекция, аттестация, обнаружение конфликтов и несоответствий, анализ взаимодействия компонентов, их функциональности и разрешение противоречий;
- целенаправленное и ориентированное на варианты использования моделирование, прототипирование и методы анализа;
- управление требованиями: отслеживание, приоритеты, изменения, базовые линии и инструментальная поддержка;
- согласование требований к компонентам и управление рисками;
- интеграция и анализ требований к компонентам, их объединения и процессов разработки комплексов программ.

Требования к квалификации специалистов по проектированию компонентов заказных программных продуктов должны использоваться и контролироваться заказчиками, менеджерами – руководителями крупных проектов, аналитиками и ведущими специалистами, обеспечивающими все этапы жизненного цикла заказных программных комплексов и систем. Они могут служить базой при ***подготовке, отборе и внедрении систем качества предприятий***, создающих сложные конкурентоспособные программные продукты.

Во всякой иерархии специалисты имеют тенденцию достигать ***своего уровня некомпетентности***. Если человек успешно справляется со своими обязанностями, его считают подходящей кандидатурой для повышения статуса. После ряда выдвижений он достигает уровня, где обнаруживается его некомпетентность, так как его но-

вые обязанности оказываются ему не по силам. Больше его не повышают, но он остается на том месте, куда попал, хотя с обязанностями своими по-прежнему справиться не в состоянии. Каждая разновидность способностей по-своему вполне реальна, но трудно сравнима с компетентностью, предполагаемой у претендентов на руководящую должность. Этот процесс приводит к тому, что многие, особенно ***руководящие должности заняты некомпетентными людь***ми, долго остающимися на своих постах. Целесообразно отказываться от всякого повышения в должности, пока специалист еще находится на уровне своей компетентности. Большинство повышений определяются компетентностью кандидата, обнаруженной им на низшей ступени деловой лестницы.

Глава 1.3

ПРОЕКТИРОВАНИЕ ТРЕБОВАНИЙ К КОМПОНЕНТАМ И КОМПЛЕКСАМ ПРОГРАММ

Общие требования к проектированию сложных программных продуктов

Команда проектирования заказного программного продукта должна ***понять проблемы заказчика до начала производства программного продукта***. Для этого следует использовать ***анализ, выявление и освоение его профессиональных проблем и интересов*** – рис. 1.5. Программный менеджер и/или системный инженер должен быть знаком с основными способами проектирования сложных систем, знать, ***как перевести расплывчатые требования и пожелания заказчика системы в четкое техническое задание***, уметь разговаривать с заказчиком и потенциальными пользователями системы на языке предметной области. ***Понимание потребностей заказчика и пользователей*** необходимый организационный этап, так как разработчики редко получают совершенные спецификации требований к создаваемой системе, они должны сами ***добывать у заказчика*** информацию, необходимую им для успешной работы. Термин ***выявление требований*** точно отражает данный процесс, в котором проектировщики должны играть активную роль. В соответствии с принятой политикой предприятия для ***определения требований к системе и продукту*** должны осуществляться ***следующие действия заинтересованных специалистов проектирования*** [1, 5, 14]:

- идентификация заинтересованных лиц или групп, имеющих законный интерес к системе и программному продукту в течение жизненного цикла;
- определение ограничений системных решений, которые являются неизбежным следствием существующих соглашений, управленческих или технических решений специалистов;

Проектирование общих требований к компонентам и комплексам программ включает:

- общие требования к проектированию сложных программных продуктов;
  - анализ, выявление и освоение профессиональных проблем заказчика;
  - этапы проектирования производства заказных программных продуктов;
  - особенности внешней среды системы;
  - распределение системных требований между аппаратными и программными компонентами, интерфейсов с внешней средой;
  - ограниченные ресурсы для реализации требований функциональной пригодности, время и допустимая длительность проектирования;
- общие требования и ограничения системы и комплекса программ реального времени;
  - три пространства функций и характеристик – заказчика, проектировщика, пользователя;
  - функциональные требования к проектированию сложных заказных комплексов программ;
  - выявление и прояснение нечетких, неоднозначных требований;
  - анализ содержания требований на значимость.
- формирование требований компонентов и модулей путем декомпозиции функций комплексов программ:
  - унификацию архитектуры и интерфейсов модулей, компонентов и комплексов программ;
  - нисходящее - восходящее проектирование модулей и программных компонентов;
  - распределение и установление ответственности специалистов за качество и сроки создания модулей и компонентов.
- повторное использование модулей и компонентов в комплексах программ:
  - цели и требования создания и повторного применения программных компонентов и модулей;
  - подготовку возможности многократного использования компонентов в различном операционном и внешнем окружении;
  - оценивание затрат на применение готовых компонентов при проектировании комплексов программ;
  - изменения трудоемкости и длительности создания комплексов

Рис. 1.5

- установление масштаба и требуемых ресурсов для реализации системы и программного продукта;
- определение представительного набора последовательных действий специалистов для идентификации всех требуемых функциональных возможностей, которые отвечают предполагаемым сценариям и внешней среде применения, функционирования и сопровождения системы и КП;
- анализ полноты и корректности множества установленных требований к системе и КП;
- специфицирование безопасности и других требований заказчика, имеющих отношение к критическим показателям применения системы;
- разрешение проблем и конфликтов, возникающих в связи с определением требований к системе;
- документирование требований заказчика в форме, приемлемой для управления требованиями в течение проектирования и жизненного цикла системы и КП.

Цель анализа требований состоит в преобразовании потребностей заказчика, выраженных в виде пользовательского представления о системе и КП, в ***формализованные функциональные возможности***. В ходе этого процесса должно создаваться четкое представление о будущей системе, которая будет удовлетворять требованиям заказчика и не потребует специальных мероприятий в связи с ее практическим применением. В результате определяется комплекс оцениваемых требований, какими функциями и характеристиками должна обладать система и какими должны быть их значения, чтобы удовлетворить требованиям заказчика.

Определения и формирования требований заказчика состоит в формулировании требований к системе, выполнение которых должно обеспечить ***функциональные возможности***, необходимые пользователям системы и иным заинтересованным лицам, в заданной эксплуатационной среде. Должны быть определены цели создания и назначения компонентов системы, а также функции и область ее применения. Эти данные анализируются и преобразуются в общий набор требований заказчика, они описывают необходимое поведение системы в процессе взаимодействия с эксплуатационной средой, и совокупность образцовых показателей, проверка на соответствие которым является целью процесса аттестации и позволяет подтвердить, что система и КП отвечает заявленным требованиям.

Проектирование процессов производства сложных комплексов программ является ***фундаментом для обеспечения функциональной адекватности требованиям всего жизненного цикла ПК***. От полноты и тщательности проектирования зависит эффективность реализации функций системы и степень удовлетворения ожиданий и требований заказчика и пользователей. В последовательности выработки и подготовки к реализации этих требований далее учитываются ***три крупных этапа***.

- ***системное проектирование – обследование***, системный анализ существующей системы и КП, выявление их свойств и недостатков;
- ***предварительное проектирование*** – обобщение результатов системного анализа и создание предварительной ***концепции*** новой или модернизированной системы и её программного комплекса;
- ***детальное проектирование*** – разработка ***системного проекта*** производства комплекса программ и базы данных, определяющего и конкретизирующего цель, назначение, методы дальнейшего производства и всего жизненного цикла КП.

Поэтапное проектирование требований к производству способно остановить нерентабельное развитие системы и КП и избежать крупных затрат заказчикам и разработчикам. В то же время, на базе рекомендуемых при проектировании методов, инструментальных средств и стандартов может и должен быть подготовлен и обеспечен длительный, эффективный жизненный цикл производства и совершенствование версий высококачественных программных продуктов и их компонентов.

Требования заказчика при проектировании могут выражаться ***в форме потребностей, пожеланий и ограничений***. Сценарии применения системы должны использоваться для анализа ее функционирования в заданной среде, с целью ***выявления требований***, которые формально могли быть не заданы заказчиком. Также следует анализировать социальное воздействие организации на пользователей, которые могут повлиять на использование системы или сдерживать процесс ее проектирования. Стандарты и правила должны использоваться для ***определения особенностей внешней среды системы.***

Анализ корректности сформированных требований к системе и программному продукту включает выявление и идентификацию противоречивых, пропущенных, неполных, неоднозначных, нелогичных или непроверяемых требований; расстановку приоритетов

и разрешение проблем, возникающие в связи с определением требований (см. рис. 1.5). Сюда же относятся требования, которые не могут быть реализованы или которые нецелесообразно реализовывать. Необходимо достигать соглашения совместно с заинтересованными лицами, по решениям, касающимся противоречивых, нецелесообразных и неосуществимых требований, устанавливать, чтобы их требования были корректно откорректированы.

Исходные данные и требования к проектированию программного продукта, включая установленные законодательные и регламентирующие нормативные требования, должны быть, оформлены документально, а их выбор проанализирован поставщиком на адекватность. Спецификацию требований должен представить ***потребитель – заказчик***. Однако по взаимному согласию ее может подготовить ***поставщик – разработчик***, в тесном сотрудничестве с потребителем для предупреждений разногласий путем, уточнения определений терминов, объяснения предпосылок и обоснования требований. Неполные, двусмысленные или противоречивые требования должны быть предметом урегулирования с заказчиком и заинтересованными лицами, ответственными за их предъявление.

Для конкретного комплекса программ доминирующие требования выделяются и определяются его ***функциональным назначением***. Программы для компьютеров как ***объекты проектирования***, производства, испытаний и оценки качества характеризуются в жизненном цикле следующими ***обобщенными характеристиками***:

- проблемно – ориентированной областью применения, техническим и социальным назначением программного комплекса;
- конкретным классом и назначением решаемых функциональных задач с достаточно определенной областью применения квалифицированными пользователями;
- масштабом и сложностью комплекса программ и базы данных, решающих единую целевую задачу системы;
- архитектурой комплекса программ и базы данных;
- необходимыми составом и требуемыми значениями характеристик качества функционирования программ и величиной допустимого ущерба – риска из-за недостаточного их качества или ресурсов;
- составом потребителей характеристик комплекса программ, для которых важны соответствующие атрибуты качества;

- комплектом стандартов и их содержания, которые целесообразно использовать при выборе технологии и характеристик комплекса программ;
- реальными ограничениями всех видов ресурсов проекта;
- степенью связи решаемых задач с реальным масштабом времени или допустимой длительностью ожидания результатов решения задач;
- прогнозируемыми значениями длительности эксплуатации и перспективой создания, множества версий программного продукта;
- предполагаемым тиражом производства и применения программного продукта;
- степенью необходимой документированности программного продукта.

Исходные требования к комплексу программ могут быть представлены и согласованы в ***составе спецификации всей системы***. Если программный продукт нуждается во взаимодействии с другими программными или аппаратными продуктами, то в спецификации требований должны быть оговорены непосредственно или при помощи ссылок интерфейсы между разрабатываемыми и другими применяемыми продуктами. В этом случае должны быть разработаны процедуры, обеспечивающие четкое ***распределение системных требований*** между аппаратными и программными компонентами, а также соответствующие спецификации интерфейсов с внешней средой. При ***заключении контракта спецификация требований*** может быть определена не полностью, она может быть доработана в ходе реализации проекта.

В соответствии с принципиальными особенностями конкретного комплекса программ при проектировании должны выбираться номенклатура и значения показателей качества, необходимых для его эффективного применения пользователями, которые отражаются в технической документации и в спецификациях требований на конечный программный продукт. При ***проектировании рекомендуется набор критериев*** качества требований к комплексам программ, который включает:

- корректность – отсутствуют дефекты и ошибки в формулировках требований к комплексу программ;
- недвусмысленность – каждое требование должно быть однозначно и не допускать различного понимания и толкования специалистами;

- полнота – состав и содержание требований должны быть достаточны для производства и применения корректного комплекса программ и компонентов;
- непротиворечивость – между разными требованиями к компонентам и комплексу программ отсутствуют конфликты и противоречия;
- модифицируемость – каждое требование допускает возможность его простого и согласованного изменения и развития при производстве комплекса программ;
- трассируемость – требование имеет однозначный идентификатор и возможность детализации и перехода к производству компонента или комплекса программ.

Системная эффективность применения программных продуктов в стандартах **ISO 9126, ISO 25000** определяется степенью удовлетворения потребностей заинтересованных лиц – заказчиков и/или пользователей, которые, в ряде случаев, желательно измерять экономическими категориями: прибылью, стоимостью, трудоемкостью, предотвращенным ущербом, длительностью применения и т.п. В стандартах эта эффективность отражается основной, обобщенной характеристикой качества – ***функциональной пригодностью***. В связи с тем, что ее абсолютную величину обычно трудно измерить непосредственно и количественно, то по ряду показателей необходима и возможна качественная оценка свойств и достоинств при применении программного продукта [13, 21, 29].

Улучшение каждой, ***нефункциональной – конструктивной характеристики качества***, требует некоторых затрат ресурсов, которые в той или иной степени должны отражаться на улучшении основной характеристике качества – на функциональной пригодности. Требования к конструктивным характеристикам имеют значение для проекта постольку, поскольку они обеспечивают требуемое качество реализации основного назначения и функций комплекса программ. Поэтому для каждого проекта необходимо ранжировать характеристики и их атрибуты (приоритеты) и выделять, прежде всего, те требования, которые могут в наибольшей степени ***улучшить функциональную пригодность*** для конкретных целей.

Ограниченные ресурсы для реализации требований функциональной пригодности, могут негативно отражаться на конструктивных характеристиках: на надежности, безопасности, пропускной способности, качестве взаимодействия с внешней средой и с пользовате-

лями, качестве документации и других эксплуатационных факторах. Наиболее общим видом ресурсов, используемых в жизненном цикле комплексов программ, являются ***допустимые финансово-экономические, бюджетные затраты***. При анализе требований качества этот показатель может применяться или как вид ресурсных ограничений, или как оптимизируемый критерий. При этом необходимо также учитывать затраты на разработку, закупку и эксплуатацию системы обеспечения качества, технологии и комплекса средств автоматизации разработки программ и баз данных, которые могут составлять существенную часть совокупной стоимости программного продукта.

Каждое требование к характеристикам качества и к затратам ресурсов при проектировании первоначально обычно ***анализируются независимо***, что может использоваться в качестве исходных данных для их сопоставления с отдельными характеристиками аналогичных комплексов программ, или для представления, как составляющей вектора в многомерном пространстве требований стандартизированных характеристик качества. Обычно заказчики и разработчики первоначально устанавливают требования к каждой характеристике без учета относительных затрат на их достижение, а также без детального анализа их совместного влияния на полную функциональную пригодность у потребителей. Это может приводить к ***несбалансированным значениям требований*** к отдельным, взаимосвязанным характеристикам качества, на которые не рационально используются ограниченные ресурсы проекта, или к не адекватно низким их значениям. Требования к характеристикам комплексов программ, определяющие их функциональную пригодность, ***принципиально различаются на две группы***:

- ***требования к количественным, измеряемым, функциональным характеристикам***, непосредственно влияющим на оперативное функционирование и возможность применения программного продукта в системе, в которые входят требования к надежности, безопасности, производительности, допустимым рискам применения;
- ***требования к структурным характеристикам***, определяющим архитектуру комплекса программ, влияющие на возможности его модификации и сопровождения версий, на мобильность и переносимость на различные платформы, на документированность, удобство практического освоения и применения программного продукта вне оперативного функционирования.

Общие требования к заказным системам и комплексам программ реального времени для автоматизации управления и обработки информации о динамических объектах состоят в следующем:

- в процессах управления решением задач и вычислениями на компьютерах должно использоваться ***единое глобальное реальное время*** систем управления и обработки информации динамических объектов, а также внешней среды;
- для обработки потоки данных из внешней среды могут быть независимыми, несинхронными, различными по интенсивности, содержанию, реальному времени формирования и поступления для обработки;
- информация в сообщениях от источников и объектов внешней среды должна содержать реальное время, к которому относятся сообщения, их координаты и характеристики состояния;
- реальное время решения различных функциональных задач должно эффективно упорядочиваться в соответствии с установленной дисциплиной диспетчеризации, определенной при производстве системы, и реальным временем приема информации из внешней среды;
- алгоритмы и программы функциональных задач системы могут различаться по длительности решения и важности для пользователей, должны включать и использовать текущее и расчетное реальное время результатов решения и исходной информации;
- для эффективного использования ограниченных ресурсов производительности и оперативной памяти компьютеров целесообразно применять приоритеты и прерывания исполнения программ в соответствии с выбранными дисциплинами решения различных функциональных задач;
- информация и сообщения для потребителей и внешней среды могут выдаваться асинхронно, в соответствии с установленной дисциплиной и содержать значения реального время, которому соответствуют данные в сообщениях.

Приступая к проектированию сложного комплекса программ, необходимо произвести реалистичную оценку ресурсов проекта, выделенного времени и поставленных целей. Эти факторы совместно определяют ***базовый масштаб проекта.*** Однако не всегда увеличение ресурсов позволяет сократить время создания сложного комплекса программ. Для решения этой проблемы необходимо ***управление масштабом*** проекта комплекса программ и согласование всех изменений разработчиков и заказчиков.

Время или допустимая длительность разработки является невосполнимым ресурсом. Этот ресурс все больше определяет требования к качеству комплексов программ в процессе их производства и сопровождения. Жесткие требования заказчиков к срокам реализации проектов, естественно, ограничивают разработчиков и испытателей. Увеличение числа привлекаемых для этого специалистов только в некоторых пределах позволяет ускорять разработку. Радикальный способ увеличения реальных затрат на разработку и испытания в ограниченное время состоит в систематизации, планировании и автоматизации на всех этапах жизненного цикла комплекса программ. Сокращение масштаба комплекса до размеров, соответствующего имеющимся времени и ресурсам, может привести к конфликтам между командой разработчиков и заказчиками, потребности которых необходимо удовлетворить.

Особенности требований к сложным заказным комплексам программ реального времени

После того как базовый ***уровень масштаба продукта согласован и задан, он представляет собой основу***, вокруг которой концентрируется множество видов деятельности проектирования. Функции базового уровня могут использоваться для того, чтобы реалистически оценивать прогресс в развитии проекта. Исходя из достигнутого соглашения по отношению к базовому уровню, можно манипулировать ресурсами. Базовые функции нужно проектировать наиболее детально, тем самым подготавливая их к производству программного кода. Полезно применять трассировку потребностей пользователя к функциям эталонного уровня требований.

Изменения являются неотъемлемой частью любой разработки. Базовый уровень функций обеспечивает удобный механизм управления высокоуровневыми изменениями. ***Официальное изменение*** – это когда заказчик запрашивает новую функцию системы, которая не является частью базового уровня. Если команда проекта изначально тщательно определила базовый уровень, то следует исходить из предположения, что ***любое изменение в базовом уровне повлияет на ресурсы, график или набор функций***, которые должны быть представлены в данной версии.

Требования к программному продукту при проектировании должны детализировать и конкретизировать описания функций до уровня, позволяющего разработчикам вести производство модулей, компонентов и всего комплекса, которые могут быть проверены на

корректность их реализации. Функции и основные характеристики сложных комплексов программ условно можно отразить многомерными ***пространствами свойств и значений***, контроля и обеспечения их реализации. Особенности, соответствие и покрытие этих многомерных пространств, их взаимодействия при различных задачах применения упрощенно можно представить как три пространства, ***функции и характеристики*** которых, используются и взаимодействуют в следующих целях:

- исходные, ***утвержденные требования*** к функциям и характеристикам программного комплекса, согласованные с разработчиками в виде конкретных документов, в соответствии с которыми разработчики обязаны создать и обеспечить применение программного продукта пользователями или в составе системы;
- ***реализованные*** разработчиками функции и характеристики программного комплекса, которые обычно не могут полностью и абсолютно точно соответствовать исходным эталонным требованиям, достоверно ***не известны*** заказчику, разработчикам и пользователям, и не отражены документами;
- реальные функции и характеристики программного комплекса, которые ***практически используются*** пользователями и/или системой в соответствии с эксплуатационной документацией, и могут не совпадать с исходными реализованными требованиями, вследствие превышения их значений или не полного их использования.

В ***требованиях к продукту и процессу*** должно проводиться разграничение как свойств продукта, который необходимо получить, и процесса, с помощью которого продукт будет создаваться. При этом ряд требований может быть изложен неявно и программные требования могут порождать требования к процессу. Они устанавливают основные соглашения между пользователями (заказчиками) и разработчиками в отношении того, ***что должна делать система*** и чего от нее не следует ожидать. В любом случае, задача состоит в том, чтобы программные требования были ясны, связи между ними прозрачны, а содержание спецификаций не допускало разночтений и интерпретаций, способных привести к созданию программного продукта, не отвечающего потребностям заинтересованных лиц.

При формировании требований к программному комплексу и компонентам менеджеры должны осуществлять согласованные действия в соответствии с принятой ***политикой и процедурами проектирования***:

- определять функциональные границы компонента или комплекса в терминах его поведения и предусмотренных свойств внешней среды;
- определять каждую функцию, которая должна быть реализована в программном комплексе и компоненте для обеспечения их корректного применения;
- определять необходимые ограничения по реализации и тестированию, обусловленные требованиями или неизбежными ограничениями системы;
- определять технические и потребительские характеристики комплекса и компонентов, позволяющие оценивать технические результаты функционирования;
- задавать нефункциональные системные требования в соответствии, с которыми определяются риски и параметры программного комплекса, связанные с критическими показателями: надежностью, безопасностью, производительностью;
- на протяжении всего жизненного цикла вести учет состояния совокупности требований вместе с их обоснованиями, изменениями, связанными решениями и допущениями.

Каждое положение при проектировании ***требований к программному комплексу и компоненту должно проверяться и верифицироваться*** для установления его корректности, полноты, непротиворечивости, совместимости с требованиями других компонентов, реализуемости, тестируемости и проверяемости при испытаниях. Требуется удостоверять, что требования являются, с одной стороны, необходимыми и достаточными для удовлетворения при применении компонента, а с другой – необходимыми и достаточными входными данными для других процессов и компонентов, в частности ***для проектирования функций и архитектуры комплекса программ***.

Для этого при производстве в общем случае должны быть выделены ***руководители и коллективы специалистов,*** которые должны планировать, утверждать, выпускать, распространять и сопровождать ***комплекты достоверных документов на программный продукт***. Они должны стимулировать разработчиков компонентов, программных комплексов и их корректировок, осуществлять непрерывное, регламентированное документирование процессов и результатов своей деятельности, а также контролировать полноту и качество документов.

Техническое задание на проект и исходные требования к производству программного продукта целесообразно формировать

при проектировании на основе общих положений программной инженерии, и должны содержать поэтапно уточняющиеся, детализирующиеся и дополняющиеся при детальном и рабочем проектировании разделы:

- общие технические требования, перечень стандартов и базовых нормативных документов для выполнения проекта;
- общие требования к программному комплексу; требования к функциям и основным характеристикам качества;
- требования к внешней среде применения комплекса;
- специальные требования к аппаратной и операционной платформам для реализации программного комплекса;
- требования к структуре, оформлению и содержанию эксплуатационной и технологической документации;
- этапы и график выполнения основных работ проекта;
- ожидаемые результаты проекта и форма их представления;
- порядок контроля исполнения проекта и приемки результатов работы.

Выходные проектные данные для производства программного комплекса должны быть документально оформлены и выражены в требованиях заказчика так, чтобы их можно было проверить и подтвердить соответствие входным проектным требованиям. Эти данные должны содержать ***критерии приемки программного продукта*** заказчиком или ссылки на них, а также идентифицировать те характеристики, которые являются критическими для его надежного и безопасного функционирования и применения. Для разработчиков особенно важно формализовать требования в документах и согласовать их с заказчиком при утверждении контракта и технического задания на проект. Требования к характеристикам, утвержденные после предварительного проектирования, могут быть закреплены в техническом задании ***как обязательные для детального проектирования и производства***.

Функциональная пригодность сложных заказных комплексов программ

В самом начале и в процессе проектирования, необходимо проверять ***качество и корректность требований они*** должны быть ***верифицируемыми***. Принято считать, что требования описаны не полностью, если для них не заданы правила – ***проверки и аттестации***, то есть, не определены способы контроля корректности и утвержде-

ния. Преимущества такого подхода состоят в том, что сводятся к минимуму дорогие переделки за счет уменьшения числа ***дефектов требований***, которые можно обнаружить или предотвратить на ранних этапах жизненного цикла проекта. Определение требований напрямую связано с процедурами проверки и утверждения (аттестации), как это сформулировано в стандарте **ISO12207:2007**.

Определения качества требований позволяют выявить и прояснить ***нечеткие, неоднозначные требования***. Такое неоднозначное требование целесообразно заменить несколькими требованиями, каждое из которых будет иметь свой собственный критерий качества. Не каждое требование может иметь четкий критерий качества, который можно использовать для проверки того, удовлетворяет ли какое-либо решение этому требованию. Добавив разъяснения в критерий качества для каждого требования, можно сделать их осязаемыми и понятными. Это первый шаг по определению критериев для измерения качества решений.

Когда формулируются требования помимо тех, которые уместны для конкретного проекта, есть вероятность, прихватить и несущественные требования, которые часто появляются в результате ***непонимания заинтересованной стороной целей проекта***. Уровень конкретизации и неоднозначности требований взаимосвязаны, и руководителям-менеджерам проекта необходимо стремиться установить баланс ***«золотую середину»*** между ними.

Необходимо уметь проверять, что разрабатываемый комплекс программ удовлетворяет каждому из зафиксированных и утвержденных требований. Нужно выделять каждое такое требование, чтобы ***проследить его развитие*** в ходе подробного анализа, проектирования и, наконец, производства программ. Каждый этап разработки комплекса программ формирует, уточняет и реорганизует требования, чтобы сделать их как можно ближе к назначению нового программного продукта. Каждое требование должно иметь ***уникальный идентификатор,*** требование должно отражать ***отдельно распознаваемую, измеряемую сущность***. В проектах сложных комплексов программ нужно применять способ работы с большим числом требований и сложными связями между ними. Следует учитывать, что наряду с существованием некоторого числа требований, связанных с одним основным событием и/или сценарием использования, любое требование может быть связано с другими событиями и/или сценариями использования.

Спецификация требований должна содержать все ***требования, которым обязан удовлетворять программный продукт***. В спецификации необходимо объективно определить все, что он должен делать, а также те условия внешней среды, при которых он должен применяться и функционировать. Наиболее ответственный аспект формирования требований – общение со специалистами, которые ***вносят и изменяют требования***. При наличии согласованного способа формирования требований все заинтересованные стороны могут принимать участие в процессе проектирования требований. Как только сформулировано хотя бы одно требование, можно приступать к его тестированию на корректность и задавать заинтересованным сторонам подробные вопросы для уточнения и конкретизации. Можно применять различные тесты для ***проверки того, что требование существенно*** и что все ответственные участники проекта понимают его смысл одинаково. Целесообразно заказчику определять относительную значимость, приоритеты требований, задавать критерий качества для каждого требования и использовать этот критерий для тестирования корректности возможных решений.

Системная эффективность – функциональная пригодность при проектировании комплекса программ может быть описана количественно или качественно, в виде набора полезных свойств и характеристик программного средства, их отличий от имеющихся у других комплексов программ, а также источников возможной эффективности. Она определяет назначение, основные функции и требования заказчика, какие задачи должны ***обязательно решаться*** для удовлетворения пользователей, а дополнительные, конструктивные характеристики качества – как и при каких условиях заданные функции могут выполняться с требуемым качеством. В результате может быть формализована ***цель использования*** и набор главных характеристик, требований заказчика и пользователей при заказе или приобретении программного продукта, а также предполагаемая его сфера назначения и применения. Полнота и точность представления этих характеристик является исходной для прослеживания реализации всех последующих, производных свойств и качества функциональной пригодности комплексов программ (см. рис. 1.5).

Функциональная пригодность – это набор и описания атрибутов, определяющих ***назначение, основные, необходимые и достаточные функции программного комплекса***, заданные техническим заданием и спецификациями требований заказчика или потенциаль-

ного пользователя. Номенклатура и значения всех остальных показателей качества непосредственно определяются требуемыми функциями программного комплекса и, в той или иной степени, влияют на выполнение этих функций. Данное требование связано с тем, ***какие основные*** функции и задачи должен решать программный продукт для удовлетворения потребностей пользователей, в то время как, конструктивные требования связаны с тем, ***как и при каких условиях***, заданные функции могут выполняться с требуемым качеством**.** Поэтому выбор функциональной пригодности, подробное и корректное описание ее свойств, являются основными исходными данными для установления требуемых значений всех остальных стандартизированных показателей качества. Атрибутами этой характеристики могут быть функциональная полнота решения заданного комплекса задач, степень покрытия функциональных требований спецификациями и их стабильность при совершенствовании, число и полнота реализуемых требований заказчика. Такими атрибутами могут быть: функциональная адекватность программ документам и декларированным требованиям, утвержденным заказчиком; степень покрытия тестами исходных требований; полнота и законченность реализации этих требований; точность выполнения требований детальных спецификаций на функциональные компоненты.

Выбор требований к характеристикам при проектировании программных комплексов начинается с ***определения исходных данных***. Для корректного выбора и установления требований к характеристикам качества, прежде всего, необходимо определить основные особенности применения системы. На основе этих данных должен формироваться общий ***набор требуемых характеристик, свойств, их мер и значений качества для определенных потребителей в жизненном цикле программного комплекса.***

Если ***масштаб проекта и сопутствующие требования заказчика превышают реальные доступные ресурсы***, в любом случае придется ограничиваться в функциях и качестве комплекса программ. Поэтому следует определять, что ***обязательно должно*** быть сделано в первой или очередной версии системы и программного продукта при имеющихся ресурсах проекта. Для этого приходится вести переговоры. ***Привлечение заказчика*** к итерационному проектированию и управлению масштабом и функциями, повышает взаимные обязательства сторон, способствует росту взаимопонимания и доверия между заказчиком и разработчиками. Имея достаточное определение

функций продукта (концепцию) и сократив масштаб проекта до разумного уровня, можно надеяться на успех проектирования программного продукта.

Декомпозиция требований, функций, процессов проектирования компонентов и комплексов программ

На ***характеристики*** проектирования программных комплексов весьма сильно оказывает влияние возможность использования ***апробированного задела*** из предыдущих реализованных проектов. Результаты системного анализа, применение функциональных и информационных моделей предметной области, формализация спецификаций требований, функциональная ***декомпозиция программ-ных комплексов*** и последовательная детализация проектов позво-ляют применять готовые технические решения в различных формах и сочетаниях. Для этого необходимо проектирование функционально законченных программных компонентов и модулей, потенциально готовых, к многократному применению в различной внешней и операционной среде, а также в различных сочетаниях их взаимо-действия в комплексах программ.

Программные компоненты и модули для производства сложных комплексов программ могут создаваться ***двумя методами***:

- в процессе системного проектирования конкретного комплекса программ и его ***последовательной декомпозиции*** на функциональные задачи и далее на небольшие ***уникальные*** программные компоненты и модули, которые могут иметь произвольную архитектуру и интерфейсы для определенного проекта;
- ***путем поиска, подбора и повторного использования*** готовых апробированных компонентов и модулей, созданных для предшествовавших проектов, с учетом возможности их эффективного использования в других комплексах программ за счет ***унифицированной архитектуры и интерфейсов***.

Декомпозиция требований и процессов проектирования при создании конкретных компонентов и программных продуктов основывается на разбиении общей цели проекта, на несколько промежуточных целей и этапов, каждый из которых также можно разделить. Этот процесс можно повторять до тех пор, пока каждая цель не станет достаточно четкой для ее полного ***функционального, представления и оценивания характеристик***, которую руководитель смо-

жет определить по размеру, сложности выполнения и необходимым ресурсам [8, 13].

Декомпозиция работ обеспечивает руководителей базой для разбиения крупных проектировочных задач на достаточно обозримые, управляемые компоненты. После того, как последние определены, их можно использовать для распределения между специалистами с учетом их квалификации, запланированной продолжительностью выполнения, датой начала и окончания работы. Декомпозицию работ также можно использовать как исходную информацию в процессе календарного планирования проекта. При этом производится упорядочивание технологических процессов для обеспечения своевременного и скоординированного решения всего комплекса задач проектирования.

Основной задачей декомпозиции комплекса программ является идентификация всех компонентов, видов и объектов проектирования, которые могут выполнять ***отдельные специалисты или небольшие группы*** при реализации проекта. Процессы проектирования сложных комплексов программ основаны на исходных принципах ***модульности***. Процессы являются модульными в том смысле, что их компоненты: ***невелики, обозримы и взаимоувязаны с другими компонентами***. Сложность и число интерфейсов между процессами и компонентами желательно сводить к минимуму. В принципе каждый процесс предназначен для реализации уникальной функции компонента (модуля) в жизненном цикле комплекса программ, может привлекать и использовать другие процессы и компоненты для выполнения более сложной специализированной функции.

Унификация и структурирование процессов декомпозиции комплексов программ оправданы, если они обеспечивают экономию времени производства и/или применения продукта. Разработка унифицированной структуры компонентов особенно целесообразна для версий программных комплексов, когда затраты могут эффективно окупаться при проектировании множества последовательных вариантов – версий продуктов. Для обеспечения эффективного проектирования и сокращения затрат целесообразно формулировать и соблюдать ряд принципов и правил ***структурного построения и повторного применения программных компонентов*** и модулей. Эти принципы и правила могут иметь некоторые особенности в различных проблемно-ориентированных областях. Однако их формализация и выполнение в конкретных проектах обеспечивают значи-

тельное ***снижение трудоемкости и длительности проектирования*** программных продуктов и их версий. Потеря гибкости архитектуры комплексов программ, некоторое возрастание ресурсов, необходимых для их реализации, обычно полностью компенсируются ***улучшением экономических характеристик*** процессов проектирования программных продуктов.

Многоуровневое, иерархическое построение сложных программных комплексов позволяет ограничивать и локализовать на каждом уровне сложности, соответствующие ему компоненты. ***Нижнему иерархическому уро***вню представления программ соответствуют программные и информационные модули (модули данных). Эти компоненты (например, 10 – 100 модулей) объединяются в группы на ***среднем уровне компонентов программ*** определенного функционального назначения с автономной целевой задачей. Несколько групп функциональных программ образует целостный функциональный ***комплекс программ высокого уровня*** определенной системы. В сложных случаях возможно создание программного комплекса из нескольких взаимодействующих функциональных комплексов программ высокого уровня. Взаимодействие компонентов в пределах уровня целесообразно максимально ограничивать, что позволяет упростить общее координирование компонентов и проводить его преимущественно по вертикали.

Нисходящее – восходящее проектирование модулей и программных компонентов занимает важное место при проектировании сложных программных продуктов. В зависимости от роли в общем технологическом процессе создания комплексов программ различаются два метода и уровня сборки: ***нисходящий*** и ***восходящий***. При нисходящем методе компоненты высокого и последующих уровней разрабатываются и интегрируются ***сверху вниз*** на основе задачи удовлетворения требований к программному продукту при проектировании, комплексирования компонентов и реализации комплекса программ. При восходящей интеграции для разработки и сборки программного продукта используются преимущественно готовые модули и компоненты нижних уровней, которые последовательно объединяются и формируются ***снизу вверх*** в программный продукт, соответствующий требованиям комплекса.

Нисходящая технология проектирования компонентов

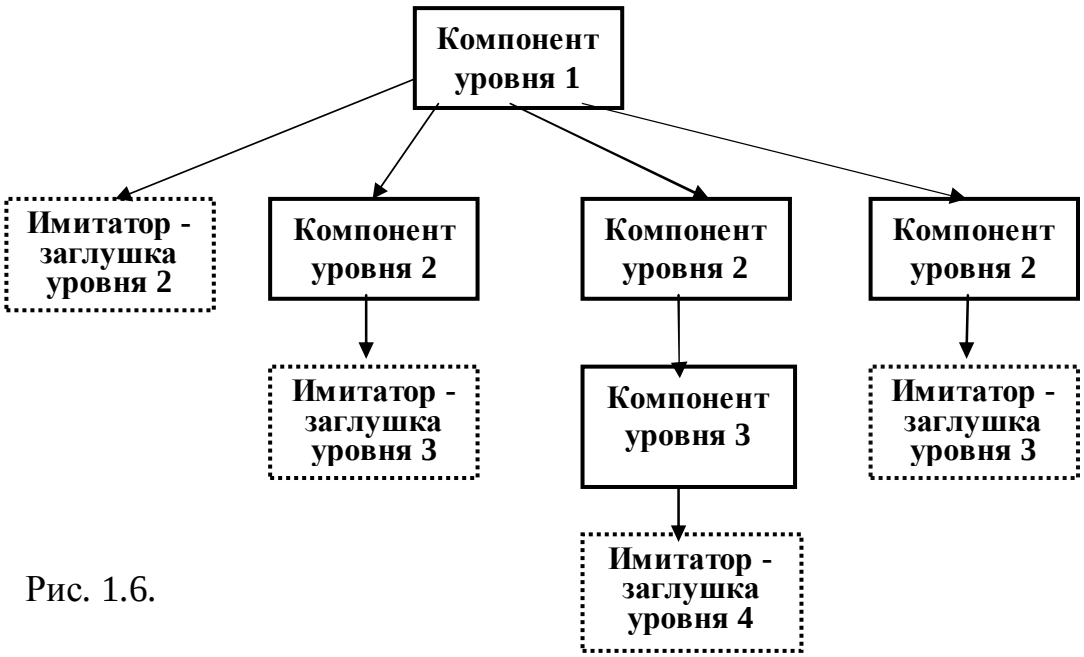

Рис. 1.6.

Восходящая технология проектирования компонентов

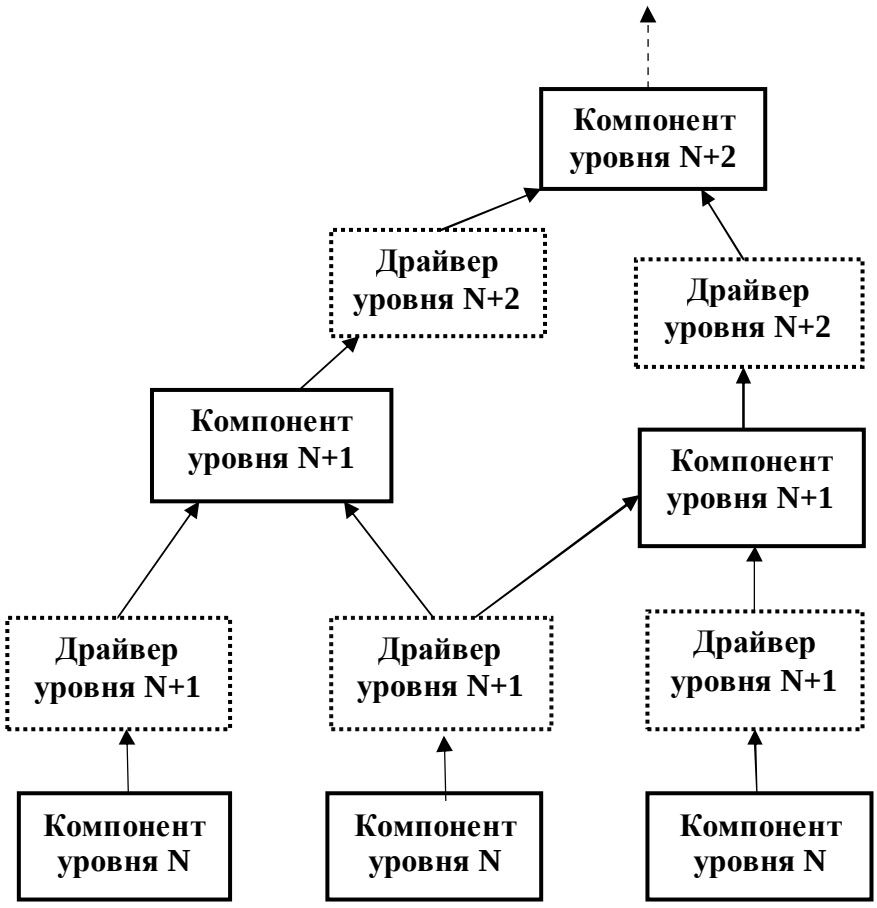

Рис. 1.7.

Нисходящий технологический процесс проектирования основан на сочетании пошаговой детализаций технического задания и требований на последовательных этапах проектирования, с ее завершением на таком уровне, когда проект может быть описан требованиями к компонентам и модулям. На основании требований технического задания и описания проблемной области создается параметризованное функциональное описание разрабатываемого комплекса программ. При проектировании осуществляется последовательная разработка спецификаций требований к создаваемому комплексу, к его составным компонентам и модулям – рис. 1.6.

При проектировании комплексов из повторно используемых компонентов (ПИК) может не оказаться доступных компонентов, полностью подходящих комплексу по функциональным и программным свойствам. В этом случае недостающие модули разрабатываются как ПИК с тем, чтобы конструктивно-технологические требования ко всем составным частям комплекса соответствовали типовой структуре, выработанной для всей проблемной области. При наличии ограничений осуществляется контроль создаваемых компонентов по функциям и конструктивным показателям (емкость памяти, интерфейсы, время функционирования).

Нисходящее проектирование – является неотъемлемой частью процесса разработки сложных программных комплексов, при котором сначала разрабатываются компоненты верхнего уровня, а затем компоненты, находящиеся на нижних уровнях иерархии. При наличии не полного состава готовых ПИК комплекс можно представить в виде одного абстрактного компонента с упрощенными, ***временными функциональными «заглушками»***, которые последовательно, по мере проектирования, заменяются на реальные модули и компоненты. Заглушки-имитаторы компонентов должны иметь такой же интерфейс, что и готовые ПИК, но может быть с ограниченной функциональностью.

Восходящий технологический процесс проектирования применяется преимущественно в тех случаях, когда разработчики хорошо знают функции, структуру и имеют почти ***весь состав готовых модулей и компонентов***. Существует ***прототип*** создаваемого комплекса, соглашения, принятые, при разработке прототипа, распространяются и на новый программный комплекс. Вначале оценивают состав и качество имеющихся готовых ПИК, из них планируют собрать предварительную версию, выбирают направ-

ления и объекты для последовательных доработок. Доработки, как правило, касаются изменения и совершенствования функциональных возможностей, интерфейсов разных видов при смене оборудования или операционной системы, изменений эргономических требований или модификации дисциплины обмена с абонентами сети или вычислительной системы – рис. 1.7. При восходящем процессе проектирования полезно осуществлять отбор компонентов, которые могут рассматриваться как повторно используемые, а также ввод их в базу данных испытанных гарантированных ПИК.

На практике при проектировании сложных программных продуктов чаще всего используется ***композиция восходящих и нисходящих методов***. Разные сроки разработки для разных компонентов предполагают, что группа, проводящая интеграцию комплекса, должна работать с какими-либо версиями готовых компонентов. Поэтому во время процесса сборки приходится разрабатывать как заглушки, так и тестовые драйверы. Завершающим этапом при нисходящем технологическом процессе является ***отбор и наполнение базы данных предприятия новыми высококачественными апробированными компонентами и модулями***. Этот этап важен не только как поставляющий новые ПИК, но и как методическая основа для обучения специалистов современным методам проектирования и интеграции комплексов программ. Такая стратегия рассчитана на несколько лет, на которую ***должно ориентироваться*** предприятие, осуществляющее проектирование программных продуктов в некоторой проблемной области.

Повторное использование готовых компонентов при проектировании программных комплексов

Требования заказчиков и пользователей по совершенствованию и снижению затрат на информатизацию объектов и процессов отразились на формировании основных ***целей и требований проектирования повторного применения программных компонентов и модулей***, которые состоят в следующем[8, 29, 34,]:

- обеспечение сохранения инвестиций, вложенных в реализованные и апробированные программные комплексы и компоненты, в процессе развития, модификации и появления новых требований к ним, а также при совершенствовании архитектур и возрастании ресурсов и функций аппаратных и операционных платформ;

- снижение трудоемкости, стоимости и длительности проектирования сложных программных продуктов;
- обеспечение возможности эффективного по экономическим показателям и качеству переноса апробированных программных компонентов и комплексов, с минимальными изменениями на различные операционные и аппаратные платформы;
- экономная реализация совместной работы и расширения функций программных продуктов во взаимодействии с другими комплексами программ при решении крупной единой целевой задачи на различных локальных и распределенных платформах;
- обеспечение взаимодействия пользователям с программными продуктами в унифицированном стиле, облегчающем им переход к использованию новых или с расширенными функциями системам.

Для этого следует проектировать технологию, а также применять стандарты, поддерживающие требования и разработку повторно используемых и/или переносимых (мобильных) программ и данных (см. стандарт **ISO 14764**). Основные особенности требований к компонентам и комплексам программ для повторного использования для проектирования в системах, ***определяют две группы задач***:

- ***структурирование*** программ и данных на стадии проектирования систем, предполагающее последовательную декомпозицию заданных функций комплекса программ, что позволяет выделять программные компоненты и модули, которые могут быть применены повторно как готовые, и описание их взаимодействия с другими компонентами;
- ***сборку или интеграцию*** готовых модулей и компонентов, а также комплексное проектирование программного продукта в целом.

Процессы переноса программ и данных на иные платформы, выбор методов обеспечения мобильности комплексов программ, и характеристики используемых ресурсов для их реализации, прежде всего, зависят от параметров компонентов, предполагаемых для переноса. Ресурсы требуются в той или иной степени на ***двух фазах реализации требований*** переноса комплексов и компонентов программ:

- при создании потенциально переносимых компонентов и комплексов программ, когда требования эффективной мобильности предусматриваются и реализуются при их проектировании, и определяются возможные платформы и области повторного применения таких программ и данных;

- при непосредственной реализации требований переноса компонентов и/или программных комплексов, в различной степени подготовленных для переноса на иные платформы и/или для повторного использования на той же платформе.

В каждом конкретном случае, ***необходима оценка рентабельности применения требований мобильности*** к компонентам с учетом ряда факторов, характеризующих комплекс программ и данных, по сравнению с полной разработкой аналогичных программных продуктов. Для достижения перечисленных целей и обеспечения мобильности требуются различные ресурсы при их проектировании. Потребность в конкретных ресурсах и рентабельность их использования зависит от ряда параметров, которые образуют широкий спектр ситуаций для анализа и применения свойства мобильности программ и данных. Такими ***ресурсами являются***:

- трудовые затраты специалистов и время проектирования требований к дополнительным интерфейсным компонентам, обеспечивающим их эффективную мобильность на определенные типы платформ;
- дополнительные ресурсы памяти и производительности вычислительных средств, необходимые для реализации и функционирования компонентов программ, обеспечивающих их высокую мобильность, например, для реализации стандартизированных интерфейсов с внешней и внутренней средой.

При переносе программного продукта свойства системы всегда несколько изменяются, что следует учитывать при анализе целесообразности переноса. Следует также учитывать при проектировании, что любой ***перенос связан с затратами***, которые требуются для:

- системного анализа рентабельности требований переноса на иную или ту же платформу и оценки технико-экономических показателей этого процесса;
- проектирования самого процесса переноса и интеграции компонентов и/или программного продукта с операционной и внешней средой на новой аппаратной платформе или в существующей среде;
- корректировки или дополнения эксплуатационной и технологической документации на перенесенный программный продукт.

Для обеспечения переноса текстов программных компонентов необходимо при их проектировании ***подготовить потенциальную возможность их последующего многократного использования*** в различном операционном и внешнем системном окружении. Для это-

го должна быть унифицирована технология разработки модулей и групп программ, унифицированы структуры межмодульных связей и технология комплексирования, стандартизирована система идентификации и специфицирования программ и данных, а также дисциплина испытаний и документирования компонентов и комплексов программ. При этом проявляются следующие ***преимущества повторного использования программных компонентов:***

- ускорение разработки – повторное использование компонентов ускоряет создание систем, так как сокращается время на их разработку и тестирование;
- повышение надежности – компоненты, повторно используемые в других системах, оказываются значительно надежнее новых компонентов, они протестированы и проверены в разных условиях работы, ошибки, допущенные при их проектировании, обнаружены и устранены при первом их применении;
- уменьшение проектных рисков – для уже существующих компонентов можно более точно прогнозировать расходы, связанные с их повторным использованием, такой прогноз – важный фактор организации комплекса программ;
- эффективное использование специалистов – часть специалистов, выполняющих одинаковую работу в разных проектах, может заниматься разработкой компонентов для их дальнейшего повторного использования, эффективно применяя накопленные ранее знания и опыт;
- стандарты интерфейса компонентов, внешней среды и пользователя, можно реализовать в стандартных компонентах, что повышает надежность систем, так как, работая со знакомым интерфейсом, пользователи совершают меньше ошибок.

Повторное использование компонентов должно быть ***систематическим, плановым и включенным во все организационные процессы проектирования предприятия***. Перед началом этапа проектирования разработчики должны выполнять поиск компонентов, подходящих для повторного использования. Требования к комплексу программ изменяются с учетом имеющихся компонентов, выбранных для повторного использования. Такой подход предполагает определенные компромиссы в реализации требований. Хотя такая система может оказаться менее эффективной, чем система, разработанная без ПИК, этот недостаток компенсируется более низкой стоимостью разработ-

ки, более высокими темпами создания системы и ее повышенной надежностью.

Унификация всегда требует некоторых ресурсов, которые в данном случае, выражается в дополнительной трудоемкости создания ПИК программ и данных, а также в увеличении необходимой памяти и производительности компьютера для их реализации. Сохранение и развитие довольно широкого спектра архитектур компьютеров, естественно привело к повторному использованию компонентов не только на однотипных платформах, но и к производству программных средств и баз данных, переносимых на различные аппаратные и операционные платформы. Таким образом, сформировались ***две технологические проблемы***:

- ***создание*** программных модулей, компонентов и баз данных, которые рентабельно повторно применять в различных проектах и/или переносить на различные операционные и аппаратные платформы;
- ***повторное использование*** и/или перенос модулей, компонентов программ и баз данных для создания из них новых программных продуктов на различных платформах.

У разработчиков сложных ***заказных комплексов программ*** зачастую возникает проблема – проводить ли проектирование и производство ***нового*** сложного программного продукта по технологии полного жизненного цикла «с нуля» или попытаться найти прототипы и ***воспользоваться готовыми апробированными решениями и компонентами***. При этом критерием экономической эффективности использования готовых компонентов в первом приближении может быть относительное число компонентов или их доля от полной трудоемкости проектирования программного продукта, которая сокращается за счет повторного испльзования апробированных ПИК. Затраты на проектирование комплекса программ при этом практически не изменяются, но сокращаются призводственные этапы программирования, тестирования и автономной отладки компонентов, а также этапы их сборки и испытаний комплекса.

Использование готовых программных компонентов позволяет сокращать или исключать из полных затрат на производство комплекса, ***затраты на программирование и тестирование компонентов и модулей***, которые зависят от относительного их числа и объема (доля ПИК) в составе общего числа компонентов в комплексе программ. Кроме того, большое число ПИК может

отражаться на сокращении затрат на этапах предварительного и детального проектирования, также положительно влиять на уменьшение затрат на сборку, испытания и документирование программного продукта. Предельный выигрыш в трудоемкости проектирования комплекса, достигается при полном наборе готовых ПИК, когда отсутствует необходимость разработки дополнительных программных компонентов.

Глава 1.4

ТРЕБОВАНИЯ К ХАРАКТЕРИСТИКАМ КАЧЕСТВА И ДОПУСТИМЫМ РИСКАМ ПРИ ПРОЕКТИРОВАНИИ ПРОЦЕССОВ ПРОИЗВОДСТВА ПРОГРАММНЫХ КОМПЛЕКСОВ

Общие требования к качеству сложных программных комплексов

Подробный состав требований к характеристикам сложных программных продуктов, регламентирован в международных стандартах **ISO 9126** и **ISO 25000**. В них рассмотрено около двадцати основных характеристик и ряд их атрибутов качества, однако, ниже при анализе выделены доминирующие и их количество несколько сокращено. Стандартизированные в этих документах характеристики качества целесообразно использовать при обосновании производства сложных программных продуктов, к которым предъявляются ***высокие требования к качеству*** функционирования и применения. ***Системная эффективность целевого применения программных продуктов*** определяется степенью и качеством удовлетворения потребностей определенных лиц – заказчиков и/или пользователей. В стандартах эта эффективность отражается основной, обобщенной характеристикой качества – ***функциональной пригодностью программного продукта***. В связи с тем, что ее абсолютную величину обычно трудно измерить непосредственно и количественно, то по ряду показателей необходима и возможна качественная оценка свойств и достоинств основных функций продукта при применении.

Высшие приоритеты в конкретных проектах, естественно, должны присваиваться свойствам и атрибутам качества, необходимым для достижения ***стратегических целей производства,*** отражающих назначение и функции применения программного продукта. Для их реализации следует выделять достаточные ресурсы. Все остальные стандартизированные характеристики программного продук-

та, должны способствовать обеспечению ***тактических целей*** реализации выбранных конструктивных требований качества при доступных ресурсах. Поэтому им могут быть присвоены более низкие приоритеты и на выполнение соответствующих требований, могут выделяться меньшие ресурсы, что, в частности, при проектировании может отражаться на не полной реализации некоторых формализованных заказчиком требований качества вследствие ограниченности ресурсов. Стандартами рекомендуется, чтобы было обеспечено измерение каждой характеристики качества при испытаниях с точностью и определенностью, достаточной для сравнений с требованиями технических заданий и спецификаций, и чтобы измерения были объективны и воспроизводимы.

Выбор требований к характеристикам качества программного продукта, естественно, должен начинаться с определения и формирования требований к его функциональной пригодности. Это наиболее ответственная, ***стратегическая задача*** начальных этапов ***проектирования производства***. Решение задачи должно быть направлено на обеспечение требуемой высокой функциональной пригодности продукта ***путем сбалансированного установления остальных характеристик качества в условиях ограниченных ресурсов***. Излишне высокие требования к отдельным атрибутам качества, требующие для реализации больших ресурсов, целесообразно снижать, если они слабо влияют на основные, функциональные характеристики. Ориентирами могут служить диапазоны изменения характеристик, границы количественных или качественных шкал которых сверху и снизу могут быть выбраны ***на основе следующих принципов***:

- предельные значения характеристик качества, должны быть ограничены сверху допустимыми или рациональными затратами ресурсов на их достижение при производстве и совершенствовании программного продукта;
- наибольшие допустимые затраты ресурсов, например труда и времени для реализации характеристик качества, должны обеспечивать функциональную пригодность применения программного продукта на достаточно высоком уровне;
- допустимые наихудшие значения отдельных характеристик качества, могут соответствовать значениям, при которых заметно начинает снижаться функциональная пригодность при применении программного продукта;

• ограниченные значения отдельных характеристик качества, не должны негативно отражаться на возможных высоких значениях других основных приоритетных характеристик.

В процессе производства комплекса программ ***атрибуты функциональной пригодности*** (см. главу 1.3) должны конкретизироваться в спецификациях требований к комплексу программ в целом и к его компонентам, ***с учетом экономических ограничений и доступных ресурсов***. Функции программного продукта реализуются в определенной аппаратной, операционной и пользовательской внешней среде системы, характеристики которых существенно влияют на функциональную пригодность. Для выполнения требуемых функций необходима ***адекватная исходная информация от объектов внешней среды***, содержание которой должно полностью обеспечивать реализацию декларированных функций. Полнота формализации номенклатуры, структуры и качества ***входной информации*** для выполнения требуемых функций, является одной из важных составляющих при определении функциональной пригодности программного продукта в соответствующей внешней среде.

Цель и функции программного продукта реализуются тогда, когда ***выходная информация*** достигает потребителей – объектов системы или операторов-пользователей, с требуемым содержанием и качеством, достаточным для обеспечения ее эффективного применения. Степень покрытия всей выходной информацией требований: целей, назначения и функций продукта для пользователей, следует рассматривать как основную ***меру качества функциональной пригодности***. Прослеживание и оценивание адекватности и полноты состава выходной информации снизу вверх к назначению программного продукта должны завершать выбор базовых характеристик качества функциональной пригодности, независимо от сферы применения системы – рис. 1.8.

Для освоения и применения широкого состава функций, свойств и атрибутов характеристик при проектировании требований и документов к программному комплексу, необходимы квалифицированные специалисты и значительный объем работ. Полностью это может быть рентабельно при создании достаточно крупных проектов.

В этих случаях ***обсуждение и согласование между заказчиком и разработчиком*** всего состава ***документов требований*** к функциям, свойствам и рациональным значениям ансамбля характеристик качества, может отражаться на функциональной пригодности про-

граммного комплекса и технико-экономических характеристиках всего проекта. При этом важно учитывать ограниченность ресурсов при выборе свойств и значений характеристик и ***необходимость компромиссов*** между ними, вследствие многочисленных связей и взаимовлияний.

Требования к характеристикам качества при проектировании процессов производства заказных программных комплексов включают:

- стандартизированные характеристики качества структуры сложных программных продуктов;
- функциональную пригодность программного продукта:

цели; назначение; задачи; основные функции КП;

- функциональные характеристики качества программных продуктов:
  - корректность;
  - **способность к взаимодействию;**
  - защищенность – безопасность;
- количественные характеристики качества программных продуктов:

  Надежность:
  - завершенность;
  - устойчивость;
  - восстанавливаемость;
  - доступность – готовность;

  Эффективность:
  - временная эффективность;
  - используемость ресурсов компьютера;
- качественные характеристики программных продуктов:

  Практичность:
  - простота использования;
  - **изучаемость;**

  Сопровождаемость:
  - **изменяемость;**
  - тестируемость;

  Мобильность:
  - адаптируемость;
  - простота инсталляции;
  - замещаемость.

Рис. 1.8

Стандартизированные характеристики качества сложных программных продуктов

В зависимости от назначения и размера программного продукта (см. главу 1.3) почти каждая из его ***характеристик качества может стать доминирующей*** или даже почти полностью определяющей функциональную пригодность программного продукта (см. **ISO 9126** и **ISO 25000)**. В наибольшей степени функциональная пригодность во многих случаях, например, в системах реального времени, зависит от ***корректности и надежности программного продукта***. Значительные трудности при создании ориентиров для выбора мер и шкал требований к характеристикам качества, проявляются при анализе корректности, способности к взаимодействию и к безопасности программного продукта (см. рис. 1.8).

Правильность – корректность: это способность программного продукта обеспечивать правильные или приемлемые результаты в соответствии с требованиями заказчика и пользователей. Эталонами для выбора требований к корректности при проектировании могут быть: верифицированные и взаимоувязанные требования к функциям комплекса, компонентов и модулей программ, а также правила их структурного построения, организация взаимодействия и интерфейсов. Эти требования при разработке должны быть прослежены сверху вниз до модулей, и использоваться как эталоны при установлении необходимой корректности соответствующих компонентов.

Требования к характеристике ***корректность*** могут представляться в виде описания двух основных свойств, которым должны соответствовать все программные компоненты и комплекс в целом. Первое требование состоит в выполнении определенной степени (%) ***прослеживаемости сверху вниз*** реализации требований технического задания и спецификации на программный продукт при последовательной детализации и верификации описаний программных компонентов вплоть до текстов и объектного кода программ. Второе требование заключается в выборе степени и стратегии покрытия тестами структуры и функций программных компонентов, совокупности маршрутов исполнения модулей и всего комплекса программ для процесса верификации и тестирования, достаточного для функционирования программного продукта с требуемым качеством и точностью результатов, при реальных ограничениях экономических ресурсов на тестирование.

Понятие ***корректной (правильной) программы*** может рассматриваться статически вне ее исполнения во времени. Корректность программы не определена вне области изменения исходных данных, ***заданных требованиями спецификации***. Степень некорректности программ определяется вероятностью попадания реальных исходных данных в пространство значений, которое задано требованиями спецификации и технического задания, однако не было проверено при тестировании и испытаниях. Значения этого показателя зависят от функциональной корректности применяемых компонентов и могут рассматриваться в зависимости от методов их достижения и оценивания: детерминировано, стохастически и в реальном времени.

Способность к взаимодействию – состоит в свойстве комплекса программ и его компонентов взаимодействовать с множеством определенных компонентов внутренней и внешней среды. При выборе и установлении при производстве способности программных и информационных компонентов к взаимодействию, ее можно оценивать объемом технологических изменений в комплексе программ, которые необходимо выполнять при дополнении или исключении некоторой функции или компонента, когда отсутствуют изменения операционной, аппаратной или пользовательской среды. С этим показателем связана корректность и унифицированность межмодульных интерфейсов, которые определяются двумя видами связей: по управлению и по информации. При этом важно учитывать возможность повторного использования апробированных компонентов и переноса их на различные платформы. Ряд общих понятий, методов и функций, которые могут рассматриваться как достаточно полная база и набор свойств и интерфейсов компонентов, обеспечивающих высокую способность к взаимодействию, в частности обобщены в ***концепции, методах и стандартах Открытых систем*** [36, 48].

Защищенность – безопасность тесно связана с особенностями функциональной пригодности каждого программного продукта реального времени [17, 32, 49]. Разработка и формирование требований к свойствам безопасности должны осуществляться на основе потребностей эффективной реализации назначения и функций продукта при различных, реальных угрозах. В процессе проектирования должны быть выявлены потенциальные предумышленные и случайные угрозы функционированию, и установлен необходимый уровень безопасности данного программного продукта.

Эффективная система защиты информации и программных продуктов подразумевает наличие совокупности организационных и технических мероприятий, направленных на предупреждение различных угроз безопасности, их выявление, локализацию и ликвидацию. Создание такой системы защиты предусматривает ***планирование и реализацию целенаправленной политики комплексного обеспечения безопасности***. Потенциальная возможность и реальные проявления информационных диверсий, а также непредумышленных, негативных воздействий заставляет в программных продуктах высокого качества разрабатывать методы и средства для обеспечения безопасности применения пользователями всего комплекса функциональных программ в течение его жизненного цикла. Их реализация в критических системах, может требовать значительных ресурсов памяти и производительности вычислительных средств. Из-за их ограниченности и других факторов априори невозможно обеспечить отсутствие проявления дефектов производства и абсолютную защиту крупных программных продуктов, вследствие чего безопасность их функционирования в системах имеет ***всегда конечное, ограниченное значение***.

Наиболее полно ***степень безопасности системы характеризуется величиной предотвращенного ущерба – риска***, возможного при проявлении дестабилизирующих факторов и реализации конкретных угроз безопасности применения продукта пользователями, а также средним временем между возможными проявлениями угроз, нарушающих безопасность [17, 32,39]. С этой позиции ***затраты экономических ресурсов*** разработчиками и заказчиками на обеспечение безопасности должны быть соизмеримыми с возможным ущербом у пользователей от ее нарушения. Поэтому реализации угроз, в ряде случаев, целесообразно характеризовать интервалами времени между их проявлениями, нарушающими безопасность применения системы, или наработкой на отказы, отражающиеся на безопасности. Это сближает понятия и характеристики степени функциональной безопасности с показателями надежности систем. Различие состоит в том, что в показателях надежности учитываются все реализации отказов, а в характеристиках безопасности следует регистрировать только те катастрофические отказы, которые отразились на проявлении значительных рисков и нарушении безопасности.

При проектировании комплексов программ целесообразно разделять вычислительные ресурсы, необходимые для непосредственного решения основных, функциональных задач, и ресурсы, требую-

щиеся для защиты и безопасного функционирования. Соотношение между этими видами ресурсов в реальных, крупных продуктах зависит от сложности и состава решаемых функциональных задач, степени их критичности и требований к безопасности всей системы. Реальные ограничения ресурсов, используемых в процессах производства, ограниченная квалификация специалистов, изменения внешней среды и требований заказчика объективно приводят к отклонениям процессов и реализации плана от предполагавшегося. В процессе формирования технического задания следует формулировать ***основные положения методологии и план последовательного повышения безопасности*** путем наращивания комплекса средств защиты, поэтапных испытаний компонентов и определения характеристик, допустимых для продолжения работ на следующих этапах.

Количественные характеристики качества сложных программных продуктов ниже сокращенно изложены в соответствии со стандартом **ISO 9126** [16, 42, 49]. Конструктивные характеристики могут быть разделены на две группы: количественные и качественные, которые различаются возможностями конкретизацией мер оценивания. Две группы стандартизированных характеристик качества программных продуктов – Надежность и Эффективность в наибольшей степени доступны количественным измерениям (см. рис. 1.8).

Надежная программа, прежде всего, должна обеспечивать достаточно низкую вероятность отказа в процессе функционирования в реальном времени. Быстрое реагирование на искажения программ, данных или вычислительного процесса и восстановление работоспособности за время меньшее, чем порог между сбоем и отказом, обеспечивают высокую надежность программ. При этом не корректная программа может функционировать абсолютно надежно. В реальных условиях по различным причинам исходные данные могут попадать в пространство значений, вызывающих сбои, не проверенные при испытаниях, а также не заданные требованиями спецификации и технического задания. Если в этих ситуациях происходит достаточно быстрое восстановление, такое что не фиксируется отказ, то такие события не влияют на основные показатели надежности – наработку на отказ и коэффициент готовности. Следовательно, надежность функционирования программ ***является понятием динамическим***, проявляющимся во времени и существенно отличается от понятия корректности программ.

При оценке реализации требований надежности регистрируются только такие искажения в процессе динамического исполнения программ, которые приводят к потере работоспособности продукта. Непредсказуемость вида, места и времени проявления дефектов в процессе эксплуатации приводит к необходимости создания специальных, дополнительных систем оперативной защиты от непредумышленных, случайных искажений вычислительного процесса, программ и данных. Системы оперативной защиты предназначены для выявления и блокирования распространения негативных последствий проявления дефектов и уменьшения их влияния на надежность функционирования комплекса программ до устранения их первичных источников. Для этого в комплекс программ должна вводиться временная, программная и информационная избыточность, осуществляющая оперативное обнаружение дефектов функционирования, их идентификацию и автоматическое восстановление (рестарт) нормального функционирования. Эффективность такой защиты зависит от используемых методов, координации их применения и выделяемых вычислительных ресурсов на их реализацию.

Основным ***принципом классификации сбоев и отказов*** в программных продуктах при отсутствии их физического разрушения, является разделение по временному показателю длительности восстановления после любого искажения программ, данных или вычислительного процесса, регистрируемого как нарушение работоспособности. При длительности восстановления, меньшей заданного порога, дефекты и аномалии при функционировании программ следует относить к ***сбоям***, а при восстановлении, превышающем по длительности пороговое значение, происходящее искажение соответствует ***отказу***. Классификация программных сбоев и отказов по длительности восстановления приводит к необходимости анализа динамических характеристик объектов, являющихся потребителями данных, обработанных комплексом программ, а также временных характеристик функционирования программ. Временная зона перерыва нормальной выдачи информации и потери работоспособности, которую следует рассматривать как зону сбоя, тем шире, чем более инертный объект находится под воздействием сообщений, подготовленных программным комплексом.

Основным ***требованием к процессу восстановления*** является ***длительность восстановления*** и ее вероятностные характеристики. Этот критерий учитывает возможность многократных отказов и вос-

становлений. Обобщение характеристик отказов и восстановлений производится в критерии ***коэффициент готовности***. Этот показатель отражает вероятность иметь восстанавливаемую систему в работоспособном состоянии в произвольный момент времени. Значение коэффициента готовности соответствует доле времени полезной работы комплекса программ и системы на достаточно большом интервале, содержащем отказы и восстановления.

Непредусмотренные при проектировании ситуации и ошибки функционирования программ и данных могут быть потенциальными источниками катастроф при применении таких программных комплексов, ***влияющими на безопасность их функционирования и применения***. Наиболее полно функциональная безопасность комплексов программ характеризуется величиной ущерба, возможного при проявлении дестабилизирующих факторов и реализации конкретных ***угроз – рисков***, а также средним временем между проявлениями непредумышленных угроз, нарушающих надежность и безопасность. Однако описать и измерить, в общем виде, возможный ущерб при нарушении безопасности для критических комплексах программ разных классов практически невозможно. Поэтому реализации угроз целесообразно характеризовать интервалами времени между их проявлениями, нарушающими безопасность применения программ, или наработкой на отказы, отражающиеся на безопасности. Это ***сближает понятия и требования функциональной безопасности с показателями надежности*** комплексов программ. Различие состоит в том, что в показателях надежности учитываются все реализации отказов, а в характеристиках функциональной безопасности следует регистрировать только те ***случайные катастрофические отказы***, которые отразились на безопасности. Статистически таких отказов может быть в несколько раз меньше, чем учитываемых в значениях надежности. Однако методы, влияющие факторы и реальные значения показателей надежности могут служить ориентирами при оценке функциональной безопасности критических программных комплексов.

Применением программно-аппаратных механизмов автоматического рестарта эта наработка при проявлении отказов, может быть значительно повышена, так как при некоторых отказах возможно их автоматическое обнаружение и оперативное восстановление работоспособности, вследствие чего значения устойчивости и наработки на отказ возрастают. Допустимая длительность прерывания оперативной

работы пользователей системы для полного восстановления нормального ее функционирования обычно может составлять несколько секунд или минут. Надежность программных продуктов наиболее полно характеризуется устойчивостью или способностью к безотказному функционированию, а также восстанавливаемостью работоспособного состояния после произошедших сбоев или отказов.

Завершенность: свойство программного продукта не попадать в состояния отказов вследствие ошибок и дефектов в программах и данных. Завершенность можно характеризовать наработкой (длительностью) на отказ (при отсутствии автоматического восстановления – рестарта), измеряемой обычно часами времени.

Устойчивость к дефектам и ошибкам: свойство программного продукта автоматически поддерживать заданный уровень качества функционирования при проявлениях дефектов и ошибок или нарушениях установленного интерфейса. Эффективное, оперативное устранение проявления дефектов, ошибок и некорректного взаимодействия с операционной и внешней средой определяют характеристику – устойчивость программного продукта.

Восстанавливаемость: свойство программного продукта в случае отказа возобновлять требуемый уровень качества функционирования, а также поврежденные программы и данные. Основными показателями качества процесса восстановления являются его длительность (минуты, секунды) и вероятностные характеристики.

Доступность или готовность: свойство программного продукта быть в состоянии выполнять требуемую функцию в данный момент времени при заданных условиях использования. Для определения этой величины измеряется относительное время работоспособного состояния комплекса программ между последовательными отказами. Обобщение характеристик отказов и восстановления производится в критерии коэффициент готовности.

Эффективность: в стандарте **ISO 9126** отражена двумя характеристиками – временной эффективностью и используемостью ресурсов компьютера, которые рекомендуется описывать, в основном количественными, атрибутами, характеризующими динамику функционирования программного продукта. Основные требования к атрибутам характеристики эффективность использования вычислительных ресурсов программным продуктом, сосредоточены на наиболее критичных показателях производительности и длительности решения функциональных задач. Используемость ресурсов памяти и произво-

дительности вычислительных средств могут устанавливаться исходя, с одной стороны, из экономической целесообразности применения наиболее дешевой, с минимальными ресурсами ЭВМ, загрузка которой будет в среднем не ниже 0,5. С другой стороны высокая загрузка (выше 0,9) может приводить к задержке или даже потере заданий при случайном, кратковременном повышении их интенсивностей, что может негативно отражаться на применимости и рисках программного продукта.

Для заказных систем реального времени особое значение имеют ***требования эффективного использования*** программным продуктом ресурсов компьютера по производительности. В стандарте **ISO 9126** они отражены ***двумя динамическими характеристиками требований*** – временной эффективностью и используемостью ресурсов компьютера, которые рекомендуется описывать, в основном количественными, атрибутами, учитывающими динамику функционирования комплексов программ. Основные требования к характеристикам эффективность использования вычислительных ресурсов сосредоточены на наиболее критичных показателях производительности и длительности решения функциональных задач (см. рис. 1.8). При этом в контракте, техническом задании и спецификации требований должны быть зафиксированы, и утверждены ***требуемые значения*** этих характеристик и их приоритетов.

Временная эффективность: свойства программного продукта, характеризующие требуемые времена отклика и обработки заданий, а также производительность решения задач с учетом количества используемых вычислительных ресурсов в установленных условиях (см. стандарт **ISO 14756**). Временная эффективность программного продукта определяется длительностью выполнения заданных функций и ожидания результатов в средних и/или наихудших случаях, с учетом приоритетов задач. Пропускная способность комплекса программ на конкретном компьютере отражается числом сообщений или заданий на решение определенных задач, обрабатываемых в единицу времени, зависящую от характеристик внешней среды.

Используемость ресурсов: степень загрузки доступных вычислительных ресурсов в течение заданного времени при выполнении функций комплекса программ в установленных условиях. Ресурсная экономичность отражается занятостью ресурсов центрального процессора, оперативной, внешней и виртуальной памяти, каналов ввода-вывода, терминалов и каналов сетей связи исполнением программ.

Потребность в производительности компьютера в процессе решения функциональных задач может значительно изменяться в зависимости от их свойств, а также от потока, состава и объема исходных данных. Степень использования ограниченной производительности компьютера в некоторых пределах не влияет на качество решения функциональных задач комплексом программ. При излишне высокой интенсивности поступления исходных данных может ***нарушаться временной баланс*** между длительностью решения полной совокупности задач комплексом программ в реальном времени, и производительностью компьютера при решении этих задач. Для формирования требований к программному продукту при подготовке технического задания и спецификаций ***следует согласовывать с заказчиком динамическую модель и характеристики внешней среды***, в которой будет применяться комплекс программ, а также динамику приема и передачи данных пользователям или системе.

Наиболее сложным при проектировании является ***задание требований эффективного, динамического использования производительности компьютера в реальном времени***. При этом должна быть определена зависимость качества решения функциональных задач от интенсивности поступающей информации различных типов. Основная задача состоит в определении вероятностей и ***рисков***, с которыми может нарушаться соответствие между потребностями в производительности для решения всей требуемой совокупности задач и реальными возможностями компьютера и других компонентов системы.

Три группы качественных (конструктивных) характеристик качества программного продукта – Практичность, Сопровождаемость и Мобильность трудно измерять количественно, и они доступны в основном ***качественным оценкам их свойств*** (см. рис. 1.8). В некоторых проектах для характеристик Сопровождаемости и Мобильности при проектировании могут доминировать ***экономические меры*** трудоемкости (человеко-часы) и длительности (часы) для процедур, обеспечивающих реализацию атрибутов этих характеристик в комплексе программ.

Практичность – применимость: свойства программного продукта, отражающие сложность его понимания, изучения и использования для квалифицированных пользователей при применении в заданных условиях. Требования к практичности и ее характеристикам – понятности и простоте использования, зависят от назначения и функ-

ций программного продукта и могут формализоваться заказчиками набором свойств, необходимых для обеспечения удобной и комфортной эксплуатации продукта.

Понятность: свойства программного продукта, обеспечивающие пользователю понимание, является ли программа пригодной для его целей, и как ее можно использовать для конкретных задач и условий применения. Понятность зависит от качества документации и субъективных впечатлений от функций и характеристик продукта.

Простота использования: возможность пользователю удобно и комфортно эксплуатировать и управлять программным продуктом. Эта характеристика учитывает физические и психологические особенности пользователей и отражает уровень контролируемости и комфортности условий эксплуатации программного продукта, возможность предотвращения ошибок пользователей.

Изучаемость: свойства программного продукта, обеспечивающие удобное освоение его применения достаточно квалифицированными пользователями. Она может определяться трудоемкостью и длительностью подготовки пользователя к полноценной эксплуатации продукта. Атрибуты изучаемости зависят от возможности предварительного обучения и от совершенствования знаний в процессе эксплуатации, от возможностей оперативной помощи и подсказки при использовании программного продукта, а также от полноты, доступности и удобства использования руководств и инструкций по эксплуатации. Изучаемость можно отражать трудоемкостью и продолжительностью изучения пользователями соответствующей квалификации, методов и инструкций применения программного продукта для полноценной эксплуатации.

Сопровождаемость: приспособленность программного продукта к модификации и изменению конфигурации (см. главы 2.6 и 2.7)). Модификации могут включать исправления, усовершенствования или адаптацию комплекса программ к изменениям во внешней среде применения, а также в требованиях и функциональных спецификациях заказчика. Трудоемкость модификаций определяется внутренними метриками качества комплекса программ, которые отражаются на внешнем качестве и качестве в использовании, а также на сложности управления конфигурациями версий программного продукта (см. стандарты **ISO 14764 и ISO 15846**).

Требования к сопровождаемости количественно можно установить для характеристик изменяемости и тестируемости ***экономиче-***

скими категориями допустимой трудоемкости и длительности реализации этих задач при некоторых средних условиях, обусловленных необходимостью устранения дефектов и усовершенствованиями функций программного продукта. Для подготовки и выполнения каждого изменения (без учета затрат времени на обнаружение и локализацию дефекта) нужно устанавливать допустимую среднюю продолжительность и суммарную трудоемкость работ специалистов при их реализации.

Мобильность: подготовленность программного продукта к переносу из одной аппаратно-операционной среды в другую. Установление требований к мобильности может быть сведено к формализации трудоемкости и длительности процессов: адаптации к новым характеристикам пользователей и внешней среды, инсталляции версий программного продукта в среде пользователей и замены крупных компонентов по требованиям заказчиков или конкретных пользователей.

Проектирование требований к допустимым рискам при производстве сложных комплексов программ

Риски – это негативные события и их последствия, отражающие потери, убытки или ущерб от процессов или продуктов, вызванные реализацией угроз при наличии уязвимости и ***снижения безопасности*** применения системы. Они проявляются при недостатках и дефектах обоснования, проектирования, производства и всего жизненного цикла комплексов программ. Это негативные последствия функционирования и/или применения программных продуктов, в результате отклонения характеристик объектов или процессов от заданных требований заказчика, согласованных с разработчиками, которые способны нарушать безопасность и вызвать ущерб системе, внешней среде или пользователю [18, 39 , 43].

Риски при случайных, дестабилизирующих воздействиях дефектов и отсутствии предумышленного негативного влияния на системы, программный продукт или информацию баз данных зависят от отказовых ситуаций, отрицательно отражающихся на работоспособности и реализации их основных функций. При этом катастрофически, критически или существенно искажается процесс функционирования программного продукта и системы, что может наносить значительный ***ущерб безопасности*** их применения. Одним из косвенных методов определения ***величины риска*** может быть ***оценка совокуп-***

ных затрат, необходимых для ликвидации негативных последствий риска в программном продукте, системе или внешней среде, проявляющихся в результате конкретного рискового события.

Анализы рисков – процессы определения источников и количественного оценивания рисков, угроз, уязвимостей, возможного ущерба, а также контрмер для их уменьшения. Для этого предварительно должны быть определены требования к характеристикам комплекса программ и оценки возможного ущерба при их нарушении. Анализ негативного воздействия, должен устанавливать критерии, используемые для идентификации вторичные последствия, распространяющиеся на компоненты и системы. Модели последствий требуются для прогнозирования размеров возможных аварий, катастроф и других негативных явлений в различных системах. Они включают идентификацию опасностей, угроз и оценки возможных последствий и ущерба от проявления рисков; проверку достоверности результатов анализа рисков; документальное обоснование возможных рисков. В результате анализа следует создавать план ***отслеживания изменения и сокращения рисков в жизненном цикле комплекса программ***, который должен регулярно рассматриваться и корректироваться.

Управление рисками – процесс идентификации, управления, устранения или уменьшения вероятности событий, которые могут негативно воздействовать на комплекс программ, систему и внешнюю среду, действия, осуществляемые для выполнения решений по мониторингу и сокращению рисков. Процесс включает анализ соотношения стоимости/эффективности контрмер, выбор, построение и испытание подсистемы ***обеспечения безопасности***, и исследование всех аспектов проявления рисков системы. Управления рисками предполагает ясное понимание специалистами внутренних и внешних причин и возможных реальных источников угроз, влияющих на качество программного продукта, которые могут привести к большому ущербу.

Необходимым требованием к специалистам для выполнения анализа и управления рисками должно быть ***достоверное знание целей и структуры комплекса программ, исследуемой системы и внешней среды***, а также доступных методов анализа и прогнозирования рисков. Область применения методов анализа и сокращения рисков должна быть определена и документально зафиксирована. Для этого следует составить описание основных проблем, определивших

целесообразность проведения исследования рисков. Для выработки плана исследований ***область анализа рисков*** должна быть определена, документально установлена и включать:

- описание оснований и/или проблем, повлекших необходимость анализа рисков, которое включает: формулировку задач анализа рисков, основанных на идентифицированных потенциальных опасностях, угрозах; определении критериев работоспособности и отказов системы;
- описание исследуемой системы – определение границ и областей интерфейса со смежными системами; описание условий окружающей среды;
- установление возможных источников, предоставляющих подробную информацию о всех технических, связанных с окружающей средой, правовых, организационных и человеческих факторах, имеющих отношение к анализируемым действиям и проблеме; обстоятельства, влияющие на безопасность;
- описание используемых предположений и ограничивающих условий при проведении анализа и прогнозировании рисков;
- формулировку возможных решений по сокращению рисков, которые могут быть приняты, описание требуемых выходных данных, полученных по результатам исследований и от лиц, принимающих решения.

Общей задачей анализа риска является обоснование и подготовка решений, касающихся сокращения рисков критических программных продуктов и систем на двух основных стадиях жизненного цикла.
На стадии проектирования:

- предоставление исходных данных для оценки качества системы в целом;
- выявление главных возможных источников угроз рисков и предполагаемых факторов, существенно влияющих на риски;
- определение и оценка эффективности возможных мер обеспечения безопасности, закладываемых в программный продукт и систему;
- предоставление исходных данных для оценки потенциально опасных действий, компонентов оборудования или системы;
- обеспечение специалистов соответствующей информацией при проведении опытно-конструкторских работ, ориентированных на нормальные и чрезвычайные условия функционирования комплекса программ и системы;

- оценка рисков с учетом регламентов и других требований поддержки применения комплексов программ;
- оценка альтернативных, конструктивных решений для сокращения рисков при проектировании комплекса программ.

На стадии производства и эксплуатации комплекса программ:

- контроль и оценку данных эксплуатации с целью сопоставления фактических показателей качества с соответствующими требованиями;
- обеспечение исходными данными процессов производства, методик эксплуатации, технического обслуживания, контроля и действий в чрезвычайных ситуациях проявления рисков;
- корректировку информации об основных источниках угроз, рисков и влияющих факторах;
- предоставление информации по значимости риска для принятия оперативных решений его сокращения;
- определение влияния на риски изменений в организационной структуре, производстве, процедурах эксплуатации и компонентах программного продукта и системы;
- подготовку персонала к применению программного продукта и системы при возможности проявления рисков.

Проявления рисков могут включать взаимосвязанные стоимостные и плановые виды рисков. ***Стоимостные риски*** составляют: нарушения ограничения суммарного бюджета проектирования и производства программного продукта; нарушения заданной ограниченной длительности производства компонентов и комплексов программ. ***Плановые риски*** включают: ущерб от дефектов стратегии и планирования проекта производства комплекса программ; достоверности планов, сроков и этапов жизненного цикла комплекса программ; нарушения требований и стандартов предотвращения, управления и сокращения рисков.

В проектах сложных систем, использующих программные продукты, риски могут быть обусловлены дефектами функциональных характеристик самих программ и их жизненного цикла, а также недостатками систем и внешней среды, в которой они используются. Основной ***ущерб от рисков программных продуктов*** проявляется в последствиях их применения – ***в дефектах и недостатках функционирования системы и внешней среды***. Поэтому анализ и оценка рисков комплексов программ должны быть тесно связаны с исследо-

ванием их проявления в системах, где они используются, причинами которых могут быть следующие ***виды рисков***: реализации функциональной пригодности программных продуктов; реализации конструктивных характеристик программного комплекса; ограничений ресурсов на проектирование и производство программного продукта. Результирующий ***ущерб*** в совокупности зависит от величины и вероятности проявления ***каждого вида риска***:

- реализация функциональной пригодности программных продуктов, включающей: назначение; функции; масштаб – размер; сложность программного комплекса;
- реализация конструктивных характеристик программного комплекса составляющих: корректность программ компонентов и комплекса; способность компонентов к взаимодействию; защищенность – безопасность; надежность – готовность; временную эффективность функционирования; сопровождаемость – изменяемость версий программного продукта;
- ресурсов проектирования и производства программного продукта обусловленных ограничениями: экономических и трудовых затрат; квалификации коллектива специалистов; технических, вычислительных ресурсов на проектирование, производство и функционирование программного продукта.

При анализе и управлении сокращением рисков программных продуктов целесообразно выделять наиболее характерные этапы их ЖЦ: технико-экономического обоснование проекта; разработку требований спецификаций; проектирование; кодирование; тестирование; и документирование. При обосновании и реализации комплексов программ, анализ и управление их рисками должны являться частью общей ***проблемы обеспечения высокого качества проекта, предотвращения и сокращения рисков в системе и внешней среде*** [7, 13, 35]. Эти процессы состоят в выявлении возможных негативных отклонений характеристик комплекса программ и системы от требований контракта, технического задания и спецификаций, а также в создании базы для принятия мер по минимизации таких отклонений, с учетом ограниченных ресурсов на их реализацию и других факторов. Для этого при оценивании рисков программных продуктов, их необходимо трансформировать в величины возможных ***рисков для систем, среды и пользователей***, которые являются важнейшими и определяющими при применении программных продуктов.

Обычно отсутствуют отдельные, предсказуемые факторы или

методы, способные существенно изменять основные риски процесса разработки программ. Риски в ЖЦ комплекса программ могут быть обусловлены недостатками или непредумышленными, ***негативными действиями различных лиц***, участвующих в создании или применении системы и программного продукта. Основными ***источниками непредумышленных рисков*** программных продуктов, которые могут приводить к ущербу при их разработке и применении, являются:

- ***заказчики,*** определяющие назначение и функций системы и программного продукта, которые могут задавать некорректные или нереализуемые разработчиками требования к ним, а также ограничивают выделенные и доступные для проекта ресурсы: бюджет, время, затраты на технологию и инструментальные средства;
- ***разработчики*** системы и комплекса программ, обеспечивающие реализацию его ЖЦ, могут допускать дефекты и ошибки при обосновании проекта, не выполнять согласованные с заказчиком требования к характеристикам и качеству комплекса программ, а также превышать допустимое использование выделенных ресурсов, что может отражаться на проявлении и последствиях рисков на различных технологических этапах;
- ***менеджеры и эксперты управления рисками*** – координаторы взаимодействия заказчиков и разработчиков, которые уполномочены принимать решения о необходимости их изменения, путем применения необходимых контрмер, а также о допустимости применения системы и/или программного продукта с прогнозируемыми или достигнутыми, конкретными уровнями рисков.

Источниками и причинами рисков функционирования могут быть также ***пользователи***, некомпетентно применяющие систему или программный продукт с отклонениями от требований документации по функциональной пригодности или с недопустимым использованием ресурсов при эксплуатации.

Подготовка исходных данных для анализа и управления рисками программного продукта должна ***устанавливать источники*** подробной информации о всех технических, связанных с окружающей средой, правовых, организационных и человеческих факторах, имеющих отношение к анализируемой проблеме, программному комплексу и системе, ***предположения*** о содержании, месте и условиях возможного проявления рисков. Для этого следует подготовить ***документацию***, которая должна быть однозначной – написана в стандартизированных терминах, уточняемых, если необходимо, соот-

ветствующими комментариями. Должна быть выполнена первичная идентификация возможных опасностей, угроз и предварительная оценка возможных их последствий, являющихся причиной рисков. Известные потенциальные опасности должны быть четко и точно определены и описаны.

Для обеспечения высокого качества программного продукта целесообразно формировать группы экспертов для анализа угроз и управления рисками проектирования и производства программного продукта. В процессе управления проектом значительное внимание должно уделяться прогнозированию угроз и рисков, имеющих как внешние, так и внутренние причины. Этот ***этап анализа*** включает:

- выделение возможных источников и угроз нарушения требований и ограничений ресурсов в жизненном цикле комплекса программ, определение критериев функциональной работоспособности и/или отказа системы и программного продукта вследствие проявления рисков;
- идентификацию и анализ причин, выделение категорий и возможных последствий проявления рисков функциональной пригодности и конструктивных характеристик программного продукта;
- идентификацию и анализ причин, выделение категорий и возможных последствий рисков нарушения ограничений доступных ресурсов для проекта комплекса программ.

Одной из самых распространенных причин и опасных источников рисков, являются ***ошибки при проектировании и оценке масштаба – размера*** программного комплекса. Эти ошибки, чаще всего, бывают случайными – непредумышленными, вследствие недостаточной компетентности заказчика или разработчика-поставщика. Однако в некоторых случаях в превышении значения согласованного масштаба заказанного продукта могут быть заинтересованы разработчики для получения больших ресурсов от заказчика, а уменьшенную оценку масштаба могут стремиться представить заказчики для сокращения выделяемых затрат на разработку программного продукта. Величина оцененного и согласованного между заказчиком и разработчиком допустимого масштаба комплекса программ непосредственно отражается на:

- ***бюджете и трудоемкости*** разработки и обеспечения всего жизненного цикла программного комплекса;
- ***затратах времени, сроках*** создания и всего жизненного цикла и применения комплекса программ;

- потребности в ***численности и квалификации*** специалистов для реализации проекта и создания программного продукта в соответствии с требованиями заказчика.

Это может приводить к значительным перекосам и ***несбалансированным значениям требований к отдельным, взаимосвязанным характеристикам комплекса программ***, на которые не рационально используются ограниченные ресурсы, или к не адекватно низким их значениям. В проектах это может угрожать значительным повышением стоимости и рисков и/или снижением конкурентоспособности производства создаваемого программного продукта из-за недостаточного уровня отдельных показателей качества.

На основе анализа и оценивания характеристик масштаба, набора требований, возможных затрат ресурсов и ***значений допустимых рисков следует определять***:

- целесообразно ли продолжать работы над конкретным проектом или следует его прекратить, вследствие больших рисков, недостаточных ресурсов специалистов, времени или трудоемкости (бюджета) разработки;
- при наличии достаточных ресурсов, следует ли провести маркетинговые исследования для определения рентабельности полного выполнения проекта и создания программного продукта для поставки заказчику или на рынок;
- достаточно ли полно и корректно формализованы концепция и требования к проекту, на основе которых проводились оценки затрат и рисков, или их следует откорректировать и выполнить повторный анализ с уточненными исходными данными.

Требования к оценке и сокращению допустимых рисков при проектировании вследствие недостаточных характеристик качества комплексов программ должны сопровождать весь их жизненный цикл. Для этого необходимо контролировать области возможного возникновения рисков, оценивать вероятности их проявления, виды и степень влияния угроз, которые следует минимизировать как можно раньше по мере их возникновения и обнаружения в процессах жизненного цикла комплекса программ.

Основной эффект по снижению рисков вследствие недостаточных характеристик должен достигаться на начальных этапах проектирования, когда возможно предотвращение или сокращение многих из них с минимальными затратами времени и других ресурсов. Для

этого в ***технологическом процессе проектирования производства*** необходимо использовать ***методы, которые включают***:

- оценивание вероятности каждого ***вида угроз*** характеристикам качества, потенциальной величины и вероятности их возможного негативного воздействия на каждую характеристику функциональной пригодности системы и программного комплекса;
- оценивание ***уязвимости*** и последствий дефектов каждой характеристики, и затрат ресурсов для восстановления требуемой функциональной пригодности системы и программного комплекса при проявлении рисков;
- планирование и разработку решений ***по контрмерам*** для обеспечения допустимого уровня интегрального риска функциональной пригодности и характеристик системы, в том числе, возможно, за счет ***изменения требований*** к программному комплексу, системе и/или доступных ресурсов;
- оценку ***вероятности сокращения рисков*** производства до допустимых пределов, при реализации процессов разработки и всего жизненного цикла программного комплекса с учетом доступных ресурсов.

Для управления рисками с целью минимизации и выделения, наиболее опасных из них, необходимо сопоставлять вероятности (частости) и последствия проявления, как рисков функциональных характеристик комплекса программ, так и рисков доступных ресурсов. Для каждого проекта эти виды рисков могут различно влиять на ***интегральный ущерб*** – риск комплекса программ, отражающийся на общем риске системы. Применяемые методы оценивания и анализа величин и вероятности рисков должны позволять определять приоритеты видов рисков с целью выделения соответствующей доли ***ресурсов на контрмеры*** для сбалансированного сокращения негативных последствий различных видов рисков.

Требования по управлению при проектировании рисками в жизненном цикле программных комплексов регламентированы международным стандартом **ISO 16085.** В стандарте **ISO 15504** содержится раздел, определяющий регламентирование и планирование процессов выявления и устранения совокупности различных рисков на протяжении всего жизненного цикла.

Поэтапное, иерархическое снижение интегрального риска программного комплекса при использовании выбранной стратегии может требовать ее корректировки в зависимости от достигаемого эффекта и

требуемых затрат на сокращение определенных рисков, необходимых для повышения качества, надежности и функциональной безопасности.

Глава 1.5

ПРОГНОЗИРОВАНИЕ СЛОЖНОСТИ ПРОЕКТИРОВАНИЯ ЗАКАЗНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Основные факторы, определяющие сложность заказных программных продуктов

Методы определения количественных значений основных характеристик сложности в проектах различаются, пока не стандартизированы, что вносит неопределенность в ***основные понятия и единицы оценивания сложности и количества информации в комплексах программ.*** Основной ***интеллектуальный труд специалистов*** вкладывается в разработку алгоритмов и текстов программ, поэтому желательно, чтобы выбранные единицы измерения сложности программного продукта были бы в наибольшей степени ***адекватны трудоемкости их проектирования и производства***. Кроме того, должна учитываться не только трудоемкость непосредственной подготовки текстов и программирование функциональных алгоритмов компонентов программ, но также отражалась трудоемкость их интеграции, комплексного создания и испытаний всего сложного программного продукта.

Основная сложность ***заказных программных продуктов*** приходятся на ***овеществленный***, преимущественно, ***интеллектуальный труд*** специалистов различных категорий. Труднее всего при проектировании прогнозировать характеристики сложности производства программного продукта в начале проекта, когда еще не сформировались достаточно четкие требования, назначение, функции, свойства, алгоритмы и размер программного комплекса, подлежащего проектированию и последующему производству. На базе этих требований необходимо делать общие выводы, стоит ли экономически заниматься данным проектом в дальнейшем, и на каких условиях следует заключить контракт с заказчиком на его выполнение.

Сложность процессов функционирования и применения программного продукта технических систем в реальном времени определяют:

- количество информации в программном продукте технической системы, участвующее при его функционировании и применении по назначению в единицу времени;
- количество информации, поступающее от источников на вход программного продукта технической системы в единицу времени;
- количество достоверной информации на выходе программного продукта для потребителей технической системы в единицу времени;
- требуемая функциональная надежность и безопасность применения информации программного продукта технической системы.

Для каждого проекта, выполняющего ответственные функции, при определении и реализации требований контрактов и технических заданий должны проектироваться и применяться система качества, методология и инструментальные средства, обеспечивающие ***требуемые качество обработки информации, надежность, безопасность функционирования и применения программных продуктов***. Для этого требуются энергия и ресурсы материализации информации, ее практического проектирования и использования, создания и ***сложности*** систем и программных продуктов. После завершения проектирования, программные продукты ***отчуждаются*** от коллектива их создателей и могут применяться, эпизодически модернизироваться и корректироваться. Это обусловливает необходимость оформлять достаточно сложную ***технологическую документацию***, позволяющую применять, сопровождать и развивать информацию проектов различными квалифицированными специалистами.

Общую экономическую смету, время работы над проектом и необходимые трудозатраты необходимо оценивать как можно раньше. При этом целесообразно поэтапно рассматривать ряд факторов, влияющих на экономические характеристики проектирования программного продукта. К основной группе факторов, ***отражающихся на экономических характеристиках сложности проектирования и производства программного продукта***, относятся:

- масштаб – размер комплекса программ, выраженный числом строк текста, функциональных точек или количеством программных модулей и компонентов в комплексе;

- количество обрабатываемых переменных или объем памяти, используемой для размещения базы данных;
- трудоемкость разработки комплекса программ;
- длительность разработки комплекса программ;
- число специалистов, участвующих в создании программного продукта.

Анализ предварительных экономических характеристик сложности проекта должен начинаться с прогнозирования возможного размера комплекса программ, с учетом того, что его ***достоверность может быть обусловлена следующими факторами***:

- проблема проектирования может быть недостаточно хорошо понята разработчиками и/или заказчиками из-за того, что некоторые факторы были упущены или искажены из-за предвзятого к ним отношения;
- недостаток либо полное отсутствие прототипов не позволили создать базу данных для экономических оценок сложности и их прогнозирования;
- проектирующее предприятие не располагает стандартами и методиками, с помощью которых можно выполнять производство и экономический прогноз размера – сложности проекта;
- менеджеры проектов полагают, что необходимо фиксировать требования к сложности и размеру программного продукта в начале проекта, заказчики же зачастую считают, что не стоит тратить время на разработку спецификаций требований, оценки размеров, сложности и экономических характеристик продукта;
- могут потребоваться дополнительные компоненты в комплексе программ (для интерфейсов системы, аппаратного обеспечения и внешней среды), что отражается на размерах программного продукта;
- недостаточно четкое представление об ограничениях ресурсов на уровне системы и возможностях совершенствования создаваемого программного продукта.

Имеющаяся ***статистика экономических характеристик проектирования программных продуктов различных классов*** позволила выявить основные факторы, от которых зависят их сложность (см. рис. 1.9).

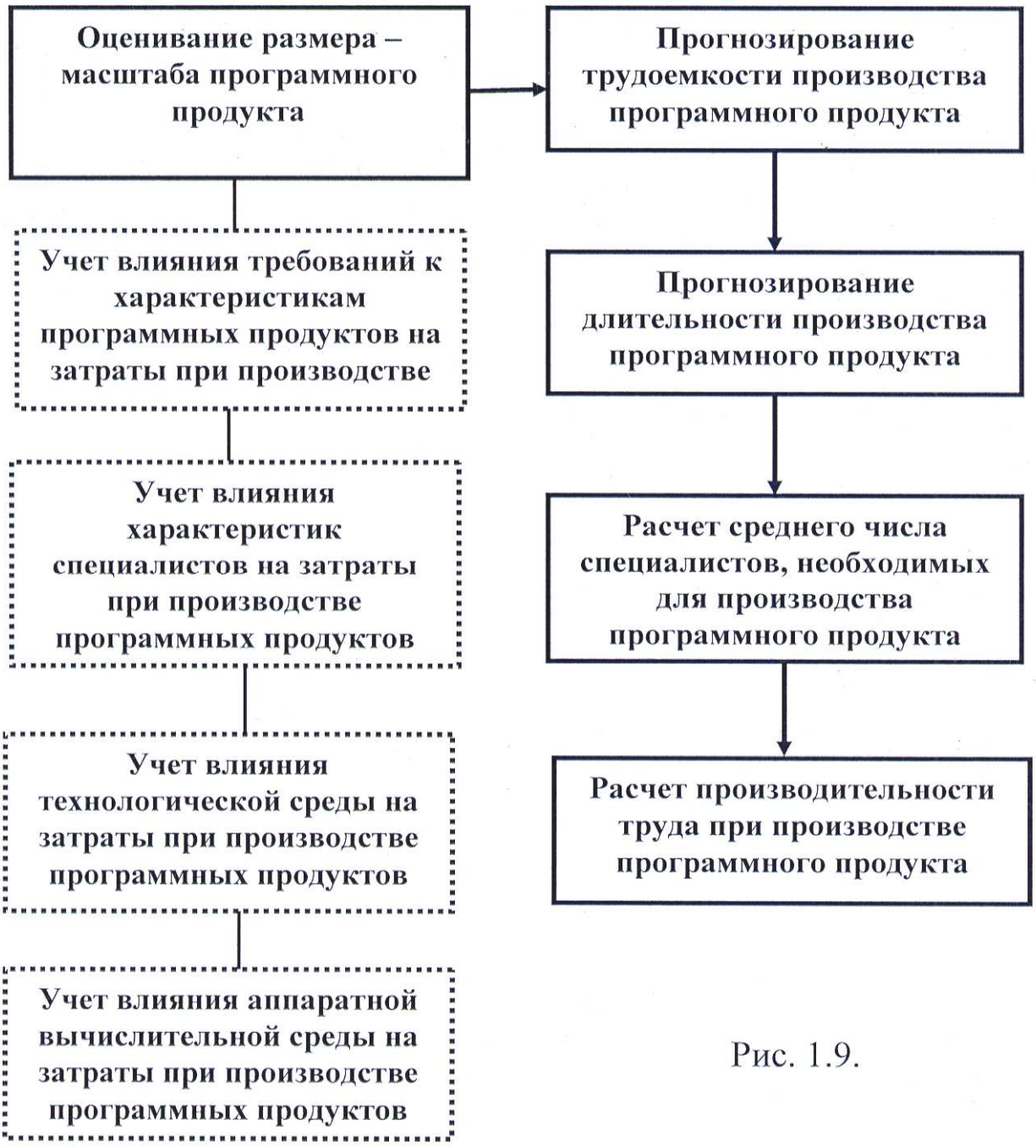

Рис. 1.9.

Изучены тенденции их изменения от важнейших параметров. Для учета классов комплексов программ с позиции экономики и сложности их производства, проведено ранжирование экспериментальных данных и выделены, ***встроенные заказные*** программные продукты систем реального времени, характеризующиеся [3, 16, 38] максимальной сложностью и трудоемкостью производства. Экспериментальные оценки трудоемкости имеют значительную дисперсию, которая обусловлена рядом трудно учитываемых факторов. Трудоемкость проектирования сложных программных продуктов размером свыше 100 тыс. строк отражается достаточно стабильными значения-

ми, что обусловлено ***усреднением творческих возможностей*** специалистов и условий их труда в больших коллективах, а также возрастанием роли руководящего и вспомогательного персонала.

Заказные программные продукты систем реального времени (СРВ) отличается от других, жесткими ограничениями на характеристики и ресурсы компьютеров, реализующих программы, и правилами использования интерфейсов. Такие продукты обычно практически полностью используют ресурсы компьютеров по памяти и производительности, снабжаются подробной документацией, эксплуатируются и поэтапно модернизируются как промышленная продукция многие годы и даже десятилетия. Эти комплексы программ непосредственно определяют степень автоматизации производства в промышленности, качество управления военными и техническими объектами и системами. Такие программные продукты должны функционировать при тесной взаимосвязи аппаратуры, программ и вычислительных процессов, а также тщательно ***оцениваться и контролироваться экономикой проектирования***.

Сложность программных продуктов зависит от многих факторов их проектирования, среди которых определяющую роль играет количество ***квалифицированного интеллектуального труда специалистов***, необходимого для ***создания и материализации информации в программах***. При этом рекомендуется учитывать, что сложные программные продукты содержат, кроме функциональной, определяющейся основными алгоритмами системы, содержат значительную часть программ, выполняющих стандартные, вспомогательные функции с малой информационной сложностью и трудоемкостью. Их следует учитывать при формализации количества и сложности информации в системах, для оценивания размеров и бюджетов создания программных продуктов при заключении контрактов между заказчиками и разработчиками на их проектирование. Ниже рассмотрены сущность ***трех методов*** оценивания и прогнозирования информационной сложности, устанавливающих ***взаимосвязь информатики и экономики*** в программных продуктах:

- ***число строк текстов программного кода*** на языке программирования высокого уровня (***Lines of code – LOC***), которое в значительной степени зависит от конкретных особенностей используемого языка, ориентированы на определенные классы задач, характеризуют ***функциональную, алгоритмическую сложность*** текстов (кодов) программ для их разработчиков;

- ***количество функциональных точек*** для измерения информационного размера и сложности программных комплексов, которые формализованы в группе международных стандартов, в них представлена концепция методов измерения программных компонентов и комплексов, перечень требований к стандартизированному методу на основе количества функциональных точек каждого программного компонента;
- результаты математических моделей оценивания ***экономических характеристик – затрат труда коллективов специалистов*** при проектировании и производстве программных продуктов, эти характеристики базируются, на экспериментальных оценках и обобщении затрат интеллектуального труда для большой совокупности (сотни) реальных завершенных программных продуктов.

Понятия и измеряемая трудоемкость организованного проектирования программных продуктов предлагается использовать как ***ориентиры*** при оценивании, сравнении и прогнозировании сложности и количества информации в технических системах. Это может быть полезно при системном проектировании и сопоставлении эффективности вариантов программных продуктов. Сравнение значений количества информации, определенных различными методами оценивания, позволяет уточнять их количество в комплексах программ и исключать грубые ошибки при прогнозировании ***экономических характеристик проектирования*** программных продуктов.

Выбор и применение единиц измерения сложности компонентов и комплексов программ одна из задач при анализе и прогнозировании ***экономики предстоящего производства*** и затрат на их создание. При одних и тех же процессах измерения продуктов производства и затрат на их создание, методики определения количественных значений основных экономических характеристик пока не стандартизированы, что вносит дополнительную дисперсию в их значения, приводимые в различных проектах и исследованиях. В результате могут делаться принципиально неверные выводы, например, об очень высокой эффективности некоторых частных технологических методов или инструментальных систем автоматизации этапов производства.

Весьма расплывчатое понятие ***сложность проектирования процессов производства комплексов программ*** значительно конкретизируется и становится измеряемым, когда устанавливается связь этого понятия ***с конкретными ресурсами, необходимыми***

для решения соответствующей задачи. Если хотя бы один из ресурсов, необходимых для решения, велик или оказывается на пределе, доступном для использования, то такую задачу вряд ли посчитают простой. Сложность задачи может быть велика по одному используемому ограниченному ресурсу и невелика из-за малого использования остальных ресурсов. Например, программа, простая по количеству модулей и длине, может оказаться сложной по объему вычислений или по числу и структуре обрабатываемых данных, и наоборот.

При производстве комплексов программ основными экономическими лимитирующими ресурсами обычно являются: допустимые трудозатраты специалистов; ограничения на сроки разработки; параметры компьютеров, технология и инструментарий проектирования и производства. Для этапа эксплуатации комплексов программ основными определяющими ресурсами может становиться память и производительность реализующего компьютера, а в качестве ограничений могут выступать время счета типовых задач, доступный объем память базы данных. Показатели сложности программного продукта можно рассматривать с позиций статической и динамической, и делить их на ***две группы***:

- ***сложность проектирования процессов производства*** программного продукта: ***статическая сложность***, когда анализируется и оценивается его назначение, функции и характеристики, а также вносятся и устраняются основные дефекты и ошибки;
- ***сложность функционирования и применения*** программного продукта для получения результатов***: динамическая сложность***, когда реализуется его функциональное назначение и качество, а также могут проявляться дефекты и ошибки.

Влияние единиц измерения размера программ на экономику проектирования производства можно значительно уменьшить, если учитывать их принципиальные особенности и разделить на ***две группы единиц измерения***:

- группу, характеризующую прогнозирование размера исходных текстов программ, которые ***разрабатываются и анализируются человеком***, отражающую сложность и трудоемкость создания комплекса программ и его компонентов;
- группу, отражающую при проектировании размер ***программ и данных, размещаемых в реализующем (объектном) компьютере***, и характеризующую его объем памяти и производительность, необхо-

димые для функционирования и исполнения программного продукта при его применении.

Эти две группы единиц отражают размер программ с разных позиций и должны использоваться в зависимости от целей анализа. Хотя между ними есть корреляция, но в общем случае размеры программ, измеренные различными группами единиц, оказываются практически несопоставимыми по экономическим характеристикам из-за наличия между ними промежуточного звена – преобразователя (транслятора) с различными, не полностью определенными характеристиками. Это обусловлено особенностями трансляторов, которые преобразуют исходные тексты программ, разработанные человеком – программистом, в разделы памяти реализующего компьютера, заполненные командами, константами или выделенные под данные, а также особенностями языков программирования, структуры и содержания машинных команд.

В системах реального времени временная сложность непосредственно отражается ***временем отклика*** (ожидания выходных результатов) на некоторые исходные данные. Пропускная способность комплекса программ при его исполнении на конкретном компьютере характеризует динамику обработки исходных данных, и сложность исполнения программного продукта в реальном масштабе времени.

Размер исходных текстов программ при проектировании отражается на трудоемкости и длительности их разработки, что позволяет оценивать относительные характеристики производительности труда специалистов – разработчиков. По большому числу проектов подтверждена ***высокая корреляция,*** между размером текста комплекса программ и трудоемкостью (с коэффициентом) его производства [3, 18, 36]. Однако исходные тексты могут содержать конструкции, которые не исполняются при рабочем функционировании программ: комментарии разных уровней, указания, заголовки и дополнительные описания, требующие определенного труда для подготовки. Для сближения понятий и значений размера программ по обеим (разрабатываемым и исполняемым) группам характеристик целесообразно эти неисполняемые части программы предварительно не учитывать.

Основной труд специалистов вкладывается в разработку текста программ, поэтому желательно, чтобы выбранная единица измерения размера – сложности была бы в наибольшей степени ***адекватна (пропорциональна) трудоемкости производства программного***

продукта. При этом должна учитываться не только трудоемкость непосредственной подготовки текста и разработки компонентов программы, но отражаться трудоемкость их интеграции и комплексного создания всего сложного продукта. Кроме того, единица измерения размера должна быть наглядной и просто измеряемой. Расчет показателей производительности труда с учетом полного размера комплекса, в том числе ранее отработанных и испытанных программ, в этом случае неправомерен, так как не учитывается труд, затраченный на готовые компоненты, используемые при сборке комплекса.

При анализе трудоемкости разработки программ, кроме затрат на создание текста программных компонентов на языке программирования, необходимо учитывать затраты: на разработку тестов для их тестирования, на комментарии к программам, на интерфейсы рзличных видов, на документацию. В результате ***сложность создания программ (непосредственный труд)*** значительно превышает, затраты труда с учетом только размера текста. ***Масштаб – размер*** текста программ может использоваться как ***исходный ориентир*** для оценки сложности производства компонентов программного продукта.

Память для программ, размещаемых в компьютере для исполнения – программная сложность, также может служить ***ориентиром*** при оценке размера комплекса программ. Он влияет на характеристики и стоимость используемых компьютеров, которая зависит от объема памяти и производительности компьютера, что особенно важно учитывать в системах реального времени (например, бортовых). Число команд, занятых программой в реализующем компьютере, соответствует числу исполняемых операторов в исходном тексте программы. За рубежом чаще всего размер комплекса программ определяется в терминах строк текста программного кода на языке программирования (Lines of code – LOC) или функциональных точек [4, 13, 14]. .

Число строк программы на языках высокого уровня (ЯВУ) в значительной степени зависит от конкретных особенностей используемого языка. Языки программирования высокого уровня ориентированы на определенные классы задач. Благодаря этому проблемно-ориентированные языки позволяют получать компактные тексты соответствующих классов программ. Однако применение их для других классов программных продуктов приводит к расширению текста и не всегда рационально. ЯВУ значительно различаются степенью обобщенного описания содержания алгоритмов, вследствие их проблем-

ной ориентации, применения разных структур операторов, операндов и процедур, что может приводить к различию (в несколько раз) числа строк текста, по сути, одних и тех же программ [3]. Число строк текста на ЯВУ характеризует сложность программы для ее разработчика, однако оно не отражает функциональную (потенциальную) сложность, которая значительно выше за счет применения процедур и комментариев. Для унификации желательно выделять ***базовый, широко применяемый язык программирования*** (например, Си) и определять коэффициенты пересчета числа строк при разработке комплексов программ на других ЯВУ с учетом классов программ.

Использование функциональных точек в качестве единиц измерения сложности программ основывается на том, что размер программного компонента (или модуля) можно оценивать в терминах ***количества и сложности функций***, реализованных в данном программном коде, а не только посредством количества строк кода [13, 29, 35]. При использовании этого метода измеряются и применяются категории пользовательских функций. При этом способ их определения является более методичным и адекватным трудоемкости, чем в случае подсчета строк LOC.

Метод функциональных точек применяется для оценки сложности функциональности программ в количественных величинах. Целью является оценка единого числа, которое бы полностью характеризовало сложность программного компонента (модуля) и адекватно соотносилось с показателями затрат труда, конкретным программистом. Фактически, функциональные точки характеризуют сложность программного компонента и могут использоваться для прогнозирования времени, необходимого на его разработку, а также нужного количества специалистов для производства программного продукта. Метод функциональных точек разработан эмпирически, используется ***в экономике производства программных комплексов*** и позволяет решать ***следующие задачи***:

- оценивать категории сложности компонентов и пользовательских функций;
- разрешать проблему, связанную с не адекватностью применения единиц измерения LOC на ранних стадиях жизненного цикла комплекса программ;
- определять и учитывать количество и сложность входных и выходных данных, запросов к базе данных, файлов либо структур

данных, а также внешних интерфейсов, связанных с программными компонентами.

Метод функциональных точек для измерения размера программных комплексов отражен в международных стандартах **ISO 20926**, **ISO 20968** и **ISO 14143:1-5**. В стандартах изложена схема оценивания выбранного метода измерения функционального размера на соответствие концепции и обязательным требованиям стандарта, для обеспечения объективности, беспристрастность и повторяемость измерений размера компонентов и комплекса программ. Схема включает требования к составу и квалификации экспертных групп, к входным документам для оценивания, к процедурам выполнения оценивания и к оформлению результатов.

Пользователи метода должны иметь возможность проверить его соответствие функциональной области проекта; оценщики методов – располагать эталонными объектами, по отношению к которым метод может быть применен и сопоставлен. Менеджеры проектов могут использовать эталонные данные для контрольного тестирования качества программного продукта и производительности при разработке проекта. Имея систему измерений функциональности компонентов, можно оценивать и сравнивать месячную ***индивидуальную производительность*** определенного специалиста или целой рабочей группы путем деления функциональности созданного программного компонента на количество месяцев, потраченное на его разработку. Это позволяет экономически более адекватно взаимодействовать производителям с заказчиком при расчетах стоимости компонентов.

Прогнозирование сложности проектирования процессов производства заказных программных продуктов на основе экономических характеристик

Основной недостаток приведенных выше методов оценивания информационной сложности и трудоемкости производства комплексов программ состоит в акценте на затраты труда только ***на создание текстов*** функциональных программ. Однако при проектировании сложных программных продуктов практически столь же большой труд вкладывается специалистами в процессы тестирования, документирования, сертификации и различные вспомогательные работы. Обобщенно все количество информации в целостном программном продукте определяется ***экономическими характеристиками – энер-***

гией труда специалистов при полном его проектировании и производстве. Эти характеристики были получены, обобщены и изучены на основе экспериментальных данных, и выявлены факторы, от которых они зависят. ***Статистические исследования экономических характеристик*** большой совокупности (сотни) реальных завершенных программных продуктов обеспечили возможность учета полных затрат интеллектуального труда, которые адекватны их информационной сложности. На основе этих данных созданы несколько математических моделей для оценивания трудоемкости, длительности и других экономических характеристик производства программных продуктов. Наиболее известной и широко используемой является модель **COCOMO II** [38, 46], основные особенности которой изложены ниже.

При увеличении размера – масштаба комплекса программ установлен более быстрый рост информационной сложности их разработки, чем отражаемый простой линейной зависимостью от масштаба функционального кода комплекса программ. Экспериментально подтверждено, что по мере увеличения базового фактора – функционального размера комплекса, возрастает относительная трудоемкость, разработки каждой строки в программе. Для учета влияния на трудоемкость основных факторов предложено использовать ***коэффициенты (рейтинги) изменения трудоемкости*** (КИТ) производства, учитывающие влияние каждого фактора совокупных затрат. В них входят факторы процесса непосредственной разработки, факторы программной и аппаратурной оснащенности, а также квалификация специалистов. Накопленный опыт и обобщение проведенных статистических исследований реальных проектов позволили детализировать влияние ***четырех основных групп факторов***, определяющих сложность экономических характеристик при проектировании производства программных продуктов – рис.1.9:

- факторы, отражающие ***требования к характеристикам функций и качества*** создаваемого комплекса программ как объекта проектирования и производства;
- факторы, определяющие ***организацию коллектива*** для проектирования и производства программного продукта и его обеспечение квалифицированными специалистами;
- факторы, характеризующие проектирование ***технологической среды*** и оснащенности инструментальными средствами автоматизации процесса производства программного продукта;

• факторы, отражающие проектирование оснащенности процесса производства программного продукта ***аппаратурными вычислительными средствами,*** на которых реализуется программный продукт и базируются инструментальные системы автоматизации разработки.

При проектировании коэффициентов в модели COCOMO II за начало разработки комплекса программ принят момент создания технического задания, а за окончание – завершение приемочных испытаний программного продукта в целом. Диапазону размеров современных программных продуктов в три-четыре порядка (до 10 млн. строк) соответствуют приблизительно такие же диапазоны изменения трудоемкости и стоимости производства программных продуктов. Однако вариации длительностей производства программных продуктов значительно меньше, чем вариация их трудоемкости, и не превышает десятикратный диапазон, от нескольких месяцев до 4 – 5 лет. Длительности разработок ограничены сверху и снизу, и одним из основных факторов, определяющих эти границы, является масштаб комплекса программ.

Для ***прогнозирования трудоемкости проектирования производства программных продуктов*** (человеко-месяцы) ***в модели COCOMO II*** [38] предложены выражения, учитывающие влияние 22 факторов в базовых значениях размера комплекса программ. В уравнении оценки трудоемкости в модели COCOMO II варьируются и применяются факторы масштабирования с целью обобщения воздействия основных параметров при прогнозировании информационной сложности производства программных продуктов. Произведение набора множителей факторов непосредственно влияет на изменение трудоемкости производства. При проектировании сложных продуктов затраты исчисляются сотнями человеко-лет, что определяет особую актуальность возможно точного учета факторов, способствующих снижению прогнозируемых затрат труда специалистов при таких масштабах производства и применения. Поэтому зачастую необходим предварительный ***системный анализ*** и тщательное распределение ресурсов на этапы производства конкретных комплексов программ путем выбора КИТ с учетом всего их жизненного цикла. При использовании выражений для прогнозирования экономических характеристик конкретных проектов следует выбирать набор факторов (***калибровать модель***), имеющих значения коэффициентов изменения тру-

доемкости (КИТ) в соответствии с характеристиками ***конкретного проекта***.

Длительность производства программных продуктов при проектировании является важнейшей экономической характеристикой, поскольку часто она определяет общие сроки разработки систем, а значит, быстроту реализации идей в различных областях. На основе значений трудоемкости, размера комплекса программ и выбранных значений факторов в модели **COCOMO II** могут быть рассчитаны ***длительность*** производства и требуемое среднее ***числа специалистов.*** При этом рекомендуется учитывать те же наборы факторов, которые применяются при прогнозировании трудоемкости непосредственного производства программного продукта.

Преимущества модели COCOMO II [3] для оценивания сложности комплексов программ состоят в следующем:

- возможен учет достаточно полной номенклатуры факторов, влияющих на оценивание экономических характеристик и сложности производства сложных программных продуктов;
- предложены таблицы для выбора возможных рейтингов для всей номенклатуры используемых факторов;
- фактические данные из таблиц подбираются в соответствии с реальными характеристиками комплексов программ и факторами калибровки, которые могут соответствовать конкретному проекту и предприятию;
- метод является достаточно универсальным и может поддерживать различные размеры, режимы и уровни качества продуктов;
- модель хорошо подходит для крупных проектов, между которыми нет существенных отличий относительно размера, сложности или функций процессов производства;
- возможна высокая степень достоверности калибровки модели с опорой на предыдущий опыт коллектива специалистов;
- результаты прогнозирования сопровождаются обязательной документацией;
- модель относительно проста в освоении и применении.

В ***системах реального времени*** их функциональная сложность непосредственно отражается на требуемом ***времени отклика*** (ожидания выходных результатов) при определенных исходных данных. Требования минимизации допустимого времени отклика (времени получения результатов) и ресурсов производительности компьютера в реальном времени может приводить к значительному повышению сложности решения задач. В этом случае задача тем

сложнее, чем ***быстрее требуется получать результаты*** при реальной ограниченной производительности вычислительных средств.

При промышленном проектировании и производстве любой продукции применяются следующие ***основные экономические характеристики***:

- ***стоимость*** – финансовые затраты на создание одного экземпляра или комплекта полноценного продукта, полностью удовлетворяющего требования пользователей;
- затраты труда – ***трудоемкость*** коллектива специалистов, участвующих в создании готового продукта, соответствующего требованиям пользователей;
- ***длительность*** – время, затраченное специалистом или коллективом на проектирование и производство продукта, удовлетворяющего пользователей;
- ***число специалистов*** разной квалификации, участвующих в создании продукта соответствующего требованиям.

Экономический анализ проектирования и производства программного продукта ***в финансовом (денежном) выражении*** имеет ряд существенных особенностей, которые ограничили его применение при оценке и прогнозировании экономики таких проектов по следующим причинам:

- предприятия, создающие комплексы программ, имеют значительные различия в уровне заработной платы специалистов, что не всегда адекватно отражается на их производительности труда и финансовых затратах;
- каждое предприятие имеет накладные расходы и налоги, которые могут значительно различаться и не влияют на трудоемкость и длительность непосредственного производства программного продукта;
- весьма различны оснащенности предприятий технологиями и средствами вычислительной техники, а также затраты на их приобретение и эксплуатацию, которые многократно используются для разных проектов;
- из общих затрат на аппаратуру и эксплуатацию технологических компьютеров и отладочных стендов сложно выделять долю, которую необходимо включать в стоимость проектирования конкретного программного продукта.

Суммарные ***затраты интеллектуального труда специалистов на проектировании программного продукта – трудоемкость***,

является основным интегральным экономическим показателем каждого программного проекта. Трудоемкость характеризуется временем труда проектирования определенного числа специалистов, необходимого для выполнения определенного этапа работ в жизненном цикле. Такой подход привел к активному использованию единиц трудоемкости: человеко-день, человеко-месяц, человеко-год (при этом человеко-год, предполагается состоящим в среднем из 250 рабочих человеко-дней с учетом выходных и праздничных дней). Эти единицы трудоемкости достаточно прочно вошли в практику проектирования и оценки процессов производства, вследствие чего являются ***базовыми в данной монографии***.

Длительность проектирования программного комплекса зависит от многих факторов и, прежде всего, от его сложности. Технологический процесс создания любых комплексов программ включает ряд базовых этапов, которые ***обязательно приходится реализовать*** независимо от затрат.

Число необходимых специалистов и особенности их квалификации неравномерно распределяется по этапам жизненного цикла комплекса программ. На начальные этапы системного проектирования и завершающие этапы испытаний программного продукта требуется относительно небольшое число, но наиболее высококвалифицированных специалистов. На средних этапах разработки и тестирования программных модулей и компонентов в проекте участвует наибольшее число специалистов относительно невысокой квалификации.

Для прогнозирования экономических характеристик новых комплексов программ при проектировании производства необходимы исходные данные:

- обобщенные характеристики использованных ресурсов и экономические характеристики завершенных разработок – прототипов проектируемого комплекса программ, а также оценки влияния на их характеристики различных факторов продуктов и среды производства;
- реализованные и обобщенные перечни выполненных работ, и реальные графики проведенного ранее проектирования программных продуктов различных классов;
- цели и содержание частных работ в процессе проектирования сложных комплексов программ и требования к их выполнению для обеспечения необходимого качества программных продуктов в целом;

- структура и содержание документов, являвшихся результатом выполнения частных проектных работ.

В средних и крупных проектах на творчество и научные исследования отдельных специалистов, (преобладающие в небольших индивидуальных проектах), накладывается множество технических работ, характерных для индустриального проектирования и производства программных продуктов. Вследствие этого значительно ***нивелировались индивидуальные особенности и квалификация отдельных специалистов***, и появлялась возможность оценивать усредненные характеристики производительности труда и другие экономические характеристики проектирования в больших коллективах.

Наиболее подробно ***основные закономерности и влияние факторов на экономические характеристики*** процессов производства сложных программных продуктов исследовались ***за рубежом.*** Эта модель была развита, детализирована и опубликована в ***2000 году*** под названием COCOMO II [38]. В этой модели на основе анализа более ***160 реальных проектов*** разработки программных продуктов различной сложности, уточнены рейтинги влияния выделенных факторов, на основные экономические характеристики проектирования производства. Обобщенные данные этих работ ниже используются и рекомендуются как базовые для прогнозирования затрат при создании сложных программных продуктов. Методическим ***примером статистических исследований*** экономических характеристик программных продуктов являются данные, полученные в ***НИР ПРОМЕТЕЙ*** **[19].** С этой целью были разработаны методические указания и анкета, разосланные в ряд предприятий для сбора сведений о каждой завершенной ***промышленной разработке программного продукта*** [15].

Характеристики трудоемкости и длительности проектирования процессов производства программных продуктов

Обеспечение функциональной пригодности является основной целью при использовании финансовых, трудовых, вычислительных и других ресурсов в жизненном цикле комплексов программ. Однако это не значит, что затраты на решение этой основной задачи всегда являются доминирующими по величине. Необходимость выполнения ряда требований к остальным, конструктивным характеристикам качества (например, надежности, безопасности), часто приводит к тому, что использование ресурсов на их реализацию может превышать ба-

зовые затраты на обеспечение только функциональной пригодности. В то же время, затраты на выполнение этих требований, всегда направлены на повышение и совершенствование функциональной пригодности. Поэтому обычно трудно четко выделить и количественно оценить все виды затраты, используемые ***только на проектирование и реализацию функциональной пригодности***.

В любом программном комплексе можно выделить компоненты, в которых сосредоточены функциональные алгоритмы и программы, предназначенные для решения основных ***целевых задач***. При проектировании затрат ресурсов в жизненном цикле комплексов программ целесообразно разделить на две части:

- затраты на создание программных компонентов, обеспечивающих ***базовые свойства функциональной пригодности*** комплекса программ для его применения по прямому назначению пользователями, в соответствии с требованиями контракта и технического задания;
- составляющие ***дополнительных затрат***, обеспечивающие требуемые конструктивные характеристики качества для улучшения функциональной пригодности продукта в соответствии с целями и сферой его применения.

Основным ресурсом при проектировании сложных комплексов программ являются ***допустимые трудозатраты*** на производство программного продукта с требуемым качеством, а также необходимое и доступное ***число специалистов*** соответствующей квалификации. Потребность в этих ресурсах в наибольшей степени зависит от размера – масштаба и сложности разрабатываемого продукта (см. главу 1.4). Когда при проектировании впервые рассматривается масштаб нового проекта, интуитивные и экспертные оценки его трудоемкости могут отличаться от конечного значения примерно в полтора – два раза в ту или другую сторону. Такая достоверность оценок обусловлена уровнем неопределенности на данном этапе знаний о конечном содержании и возможном размере программного продукта. Общая тенденция состоит в том, что на начальных этапах оценки затрат чаще всего занижаются.

Эта неопределенность уменьшается, по мере детализации и углубления содержания и функций проекта, как только фиксируются конкретные принципы функционирования и концепция комплекса программ. На этом этапе оценка достоверности размера и трудоемкости уменьшается приблизительно до 40%. Это вполне объяснимо, поскольку еще не уточнены структура и многие детали

продукта. Эти вопросы могут быть разрешены во время разработки структуры и спецификаций требований к комплексу программ и тогда можно оценить его размер с точностью до 15 – 20%. После завершения проектирования при детальном проектировании комплекса программ может быть определена структура внутренних данных и функции программных компонентов. На этом этапе ошибки оценки размера и трудоемкости проекта могут составить около 10%. ***Неопределенности оценок трудоемкости*** могут быть обусловлены: особенностями конкретных алгоритмов; управления их функционированием; обработкой ошибок; инициализацией и завершением сеансов решения задач. Уточнения размеров комплекса и компонентов могут быть решены к концу детального проектирования производства, однако при этом сохраняется неопределенность оценки размера комплекса программ и его трудоемкости порядка 5 – 10%, связанная с тем, насколько хорошо специалисты понимают спецификации требований к компонентам, в соответствии с которыми они должны создавать их программы.

Специалисты при проектировании программ ***различаются квалификацией и степенью участия*** в непосредственной разработке проекта программного комплекса. Встречаются особо талантливые программисты, способные проектировать очень быстро программные компоненты высокого качества. Однако при любых способностях ***есть предел***, доступный разработчику – одиночке, и он редко превышает 10 тыс. строк новых программ в год. Кроме того, даже в этом случае такому программисту необходима помощь при составлении тестов, оформлении документации, испытаниях и ряде других работ, обеспечивающих превращение программы в ***программный продукт***.

При проектировании сложных комплексов программ разработкой программы модулей и компонентов участвует только часть коллектива, затраты на которых необходимо учитывать при экономическом анализе. Остальные специалисты осуществляют ***руководство проектом***, проводят системный анализ, исследуют алгоритмы, оформляют документацию, выполняют вспомогательные технические работы. Если специально не оговаривается, то далее учитывается трудоемкость на программный продукт ***всех категорий специалистов коллектива*** в соответствии с их долей участия в создании конкретных программ определенного продукта.

Для учета затрат времени коллектива специалистов на проектирование конкретного комплекса программ особенно сложно фиксиро-

вать ***начало разработки***. Дело в том, что системный анализ зачастую входит в научно-исследовательские работы, финансируемые, планируемые и учитываемые независимо от начала проектирования конкретного программного комплекса. При последующем изложении началом проектирования считается создание технического задания или спецификации требований, ***согласуемых с заказчиком*** и будущими пользователями продукта.

Окончанием проектирования и производства для прекращения учета затрат при оценке трудоемкости и длительности конкретного проекта далее принимается момент завершения сдаточных или сертификационных испытаний программного продукта и оформления акта соответствующей комиссией заказчика. Однако при анализе реальных проектов эта граница оказывается тоже размытой, как и начальная, хотя и в меньшей степени. При изменении размеров сложных комплексов и трудоемкости в широком диапазоне, длительность разработки может изменяться мало по сравнению с изменением величины затрат. Для каждого размера комплекса программ при заданном качестве существует ***область невозможного сокращения длительности проектирования и производства***, которую не удается преодолеть при увеличении затрат труда. Для планирования проектирования комплекса программ и регулярного управления этим процессом необходимы частные экономические характеристики в зависимости от различных факторов. Такие показатели могут формироваться: по этапам проектирования на единицу продукции как относительные затраты на достижение заданной характеристики качества программ или как составляющие суммарных затрат в жизненном цикле комплекса программ.

Длительность проектирования производства программных продуктов является важнейшей, экономической характеристикой, поскольку часто она определяет общие сроки производства систем, а значит, быстроту реализации идей и методов в различных областях автоматизации. Диапазону размеров современных программных продуктов в три-четыре порядка (до 10 млн. строк) соответствуют приблизительно такие же диапазоны относительного изменения трудоемкостей и стоимостей их производства. Однако принципиально нерентабельно производство даже очень сложных программных продуктов более 3 – 5 лет. С другой стороны, комплексы программ даже в несколько тысяч строк по полному технологическому циклу с испытаниями и документацией как продукции, редко создаются за

время, меньшее, чем полгода – год. Таким образом, вариация длительностей проектирования производства программных продуктов меньше, чем вариация их трудоемкостей, и не превышает десятикратный диапазон. Практически длительность проектирования и производства программных продуктов ***ограничена сверху и снизу*** и одним из основных факторов, определяющим эти границы, является размер комплексов программ.

Любые прогаммные продукты должны поступать на эксплуатацию до того, как в них пропадет необходимость. Их цели, концептуальная основа и алгоритмы ***не должны устареть*** за время проектирования и производства. Отсюда появляется ***верхний предел*** допустимых длительностей проектирования. Этот верхний предел не может иметь единственное значение для любых классов и размеров программных продуктов. Однако недопустима его вариация в том же диапазоне, что размера и трудоемкости программного продукта. Поэтому на практике по мере возрастания размеров комплексов программ, увеличиваются коллективы специалистов-разработчиков, что обеспечивает основной прирост необходимой трудоемкости.

Стремление ограничивать длительность проектирования производства программных продуктов приводит к объективному формированию верхнего предела, за которым распространяется зона «***нерациональных»*** длительностей, зависящих от размера и трудоемкости комплекса программ. Даже для довольно сложных программных продуктов, имеющих размер ***свыше*** 500 тыс. строк, вряд ли допустима длительность разработки более 3 лет. Большие длительности проектирования, иногда имеющиеся на практике, обусловлены в основном низкой квалификацией разработчиков и заказчиков, недостаточной автоматизацией технологии, малым коллективом специалистов и рядом других, преимущественно организационных и технологических причин. Аналогичные ситуации чаще встречаются при относительно небольших проектах (50 – 100 тыс. строк), когда у руководителей и коллектива мал опыт их проведения, следствием чего является ***избыточный «оптимизм»*** в начале разработки, а также пренебрежение технологией и организацией работ.

Границу снизу определяют естественный технологический процесс коллективного пректирования производства и необходимость выполнения ряда скоординированных работ на последовательных этапах, которые обеспечивают получение продукта требуемого качества. В некоторых случаях увеличение числа специалистов

может давать обратный эффект – длительность производства увеличивается вместе с увеличением трудоемкости, вследствие роста затрат на взаимодействие специалистов.

Планирование этапов проектирования производства программного продукта целесообразно оценивать аддитивными экономическими показателями. Такими характеристиками могут служить суммарные трудозатраты на выполнение этапа работ при планировании и создании проекта компонентов программ определенного размера и класса или поэтапные трудозатраты на одну интегральную команду – строку текста программы. Эти характеристики позволяют выявлять наиболее трудоемкие этапы и помогают рациональнее распределять затраты по этапам работ.

Планирование проектирования сложных программных продуктов, особенно на начальных этапах, характеризуется ***высокой долей творческого труда***. Поэтому трудоемкость и длительность отдельных операций и частных работ существенно зависят от индивидуальных особенностей и квалификации их исполнителей и характеристик конкретного проекта. Отсюда принципиальной особенностью планирования проектирования комплексов программ должно являться активное участие руководителей и заказчиков проекта в прогнозировании планов на базе исследованных характеристик прототипов завершенных разработок и их личного опыта. Такие планы должны иметь разумные ограничения в детализации работ на уровне, обеспечивающем необходимую управляемость всего процесса производства.

Оценка требуемого среднего числа специалистов для проектирования и производства конкретного программного продукта предварительно может быть рассчитана путем деления оценки величины трудоемкости на длительность производства. Однако рациональное число специалистов, участвующих в проектирование распределяется не равномерно по этапам работ. При разработке программных модулей и компонентов отдельными специалистами или небольшими группами, производительность труда при написании одних и тех же текстов автономных программ, может различаться в десяток раз в зависимости от их таланта и трудоспособности, и достигать тысячи строк за человеко-месяц. При диапазоне изменения размеров программ СРВ на четыре порядка средняя производительность труда обычно изменяется только в два раза, что в ряде случаев существенно облегчает упрощенные оценки и прогнозирование экономических характеристик. Совершенствование технологии, квалификации специа-

листов и инструментальных средств автоматизации проектирования позволили повысить среднюю производительность труда при создании полностью новых оригинальных программных продуктов СРВ в несколько раз.

При проектировании комплексов программ необходимо учитывать, что экономические, временные, вычислительные и другие ресурсы ***на весь ЖЦ комплекса***, всегда ограниченны и используемые затраты для улучшения каждой характеристики должны учитывать эти ограничения. Для рационального распределения этих ресурсов необходимо знать ***как отражается изменение затрат на улучшении каждой характеристики качества программного продукта***. Эта корреляция затрат ресурсов и значений каждой характеристики качества зависит от назначения, а также от ряда свойств и других особенностей комплекса программ, что усложняет учет влияния таких связей. Тем не менее, выявлены основные тенденции такого взаимодействия, которые могут служить ***ориентирами*** при выборе и установлении требований к определенным характеристикам качества программных продуктов.

Глава 1.6

ПРОГНОЗИРОВАНИЕ ЭКОНОМИЧЕСКИХ ХАРАКТЕРИСТИК ПРОЕКТИРОВАНИЯ ПРОЦЕССОВ ПРОИЗВОДСТВА ЗАКАЗНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Простейшие модели прогнозирования экономических характеристик производства программных продуктов

При прогнозировании экономических характеристик проектирования процессов производства программного продукта, менеджерам следует учитывать цели их оценивания, потребности заказчиков и разработчиков в информации для принятия решений на этапах проектирования и производства комплекса программ. По мере разработки проекта их необходимо пересматривать и возможно изменять решения, принимаемые на этапах проектирования учитывая ***три базовых метода:***

- первичный экономический прогноз при ***подготовке концепции*** и технического задания на новый комплекс программ, на основе экспертных данных о его размере, средней производительности труда специалистов или стоимости разработки одной строки текста программ – прототипов;
- прогнозирование основных экономических характеристик ***при предварительном проектировании*** на базе экспертных значений трудоемкости и длительности разработки комплекса программ [4, 16, 46] с учетом влияния минимума дополнительных факторов;
- определение возможных экономических характеристик проекта с учетом ***доступных оценок множества факторов*** для календарного планирования производства сложного программного продукта с использованием систем прогнозирования COCOMO II [38].

Неопределенность прогнозов уменьшается, по мере детализации и углубления содержания функций проекта, как только фиксируются конкретные принципы функционирования и концеп-

ция комплекса программ. При первичном экспертном экономическом обосновании сложных комплексов программ наибольшее значение имеют ***ключевые факторы***:

- сложность и размер – масштаб, подлежащих разработке новых программных компонентов и всего комплекса (количество строк текста с указанием языка программирования или функциональных точек);
- размер и относительная доля готовых программных компонентов, которые могут быть заимствованы из предшествовавших продуктов и повторно использованы в новом проекте комплекса программ;
- относительные затраты ресурсов на создание программного продукта: труда специалистов (человеко-месяцы), времени или бюджета на единицу размера (на строку текста программ) проектируемого комплекса программ;
- возможная исходная стоимость разработки одной строки текста в комплексе программ.

Эти факторы могут быть ***прогнозируемы квалифицированными экспертами*** на основе имеющегося у них опыта реализации предшествовавших подобных проектов, а также с использованием опубликованных данных (оптимистических и/или пессимистических оценок). При наличии необходимых данных важно оценить их достоверность и возможную точность. Экспертные прогнозы предполагают использование опыта прошлых разработок и их отличия от новых методов, предусмотренных в конкретном проекте, а также учет индивидуальных возможностей специалистов коллектива разработчиков.

Экспертный прогноз удельных затрат на строку текста программного продукта обычно относится к полному циклу производства сложных комплексов программ, начиная от создания концепции и требований до завершения испытаний и передачи программного продукта заказчику или пользователям. В составе участников проекта учитываются все категории специалистов, обеспечивающие реализацию программного продукта. Уменьшение времени, затрачиваемого на весь цикл производства крупного продукта, не может быть достигнуто за счет значительного повышения производительности труда только отдельных специалистов – программистов. Она практически не зависит от усовершенствований языков программирования, организационных усилий со стороны менеджеров, от наличия или отсутствия некоторых отдельных видов инструментария и автоматиза-

ции производства компонентов и модулей. Однако ***значительно влияет доля повторно используемых компонентов***. При достаточно высоком уровне технологии большое значение имеет размер и сложность функциональных задач комплексов программ, а также ***требуемое качество и ответственность*** за создаваемые программные продукты.

Так как предприятия и разработчики сложных комплексов программ ***не заинтересованы раскрывать реальную экономику*** выполненных проектов и производства программных продуктов, опубликованные экономические характеристики программных проектов носят отрывочный, не упорядоченный и не всегда достоверный характер. Опубликованы ***ориентиры*** около 100$ и более средняя стоимость разработки одной строки текста программ реального времени высокого качества (1000$ бортовые комплексы программ системы Шатл), а для административных систем – около 20 – 50$.

Обобщенные экспертные оценки экономических характеристик проектирования и производства комплексов программ обычно представляются в виде таблиц (желательно) с указанием достоверности оценок следующих характеристик:

- полной трудоемкости производства программного продукта – ***C*** (человеко-месяцы с указанием учитываемых факторов);
- полной длительности производства программного продукта – ***T*** (месяцы);
- участвовавшего среднего числа специалистов – ***N*** (человек);
- средней производительности труда ***P*** специалистов (число строк на человеко-месяц).

Так как ***величина и достоверность определения размера комплекса программ ключевой фактор*** последующего экономического анализа, то целесообразно применять несколько доступных методов для его оценивания. Прежде всего, необходимо использовать, ревизовать и углубить данные ***оценки возможного размера продукта***, полученные при использовании предыдущей методики, с учетом повышения достоверности характеристик проекта по результатам предварительного проектирования. Конкретизация функций, структуры комплекса программ и состава компонентов проекта позволяет более достоверно определить размеры компонентов комплекса программ и, суммируя их, оценить размер всего комплекса.

В ***моделях для первичной оценки экономических характеристик*** полного цикла производства, размер комплекса программ и ос-

тальные факторы рекомендуется учитывать поправочными коэффициентами при уточнении интегральных показателей. Зависимость трудоемкости производства программного продукта ***С*** от его размеров – ***П может быть представлена простейшим выражением*** [4,15, 29]:

$$C = A \times П^{E} \times \prod_{i} M(i) \ . \qquad (1)$$

По мере увеличения размера комплекса программ возрастают не только абсолютные, но и относительные затраты на разработку каждой строки текста в программе. Вследствие этого затраты труда при разработке одной строки текста в программах разного размера могут различаться в несколько раз. При производстве комплексов программ большого размера может возрастать сложность разработки по сравнению с программами малого размера, так как в них усложняются взаимосвязи компонентов по информации и управлению, а также становятся более трудоемкими процессы планирования, тестирования и управления компонентами в ходе производства. Суммарную трудоемкость разработки сложного комплекса программ можно представить ***двумя сомножителями***. Первый сомножитель является ***доминирующим***, он прямо пропорционален размеру программы, и отражает практически линейное возрастание трудоемкости создания любых комплексов программ при увеличении их размера.

При увеличении размера комплекса программ происходит более быстрый рост сложности их разработки, чем описываемый простой линейной зависимостью. Эксперементально подтверждено, что по мере увеличения размера комплекса возрастает относительная трудоемкость разработки каждой строки в программе. Это обстоятельство можно учесть поправочным ***вторым сомножителем***, отражающим изменение трудоемкости на разработку каждой строки в программе при увеличении ее размера. Гипотеза о возрастании трудоемкости разработки с ростом размера комплекса быстрее, чем по линейному закону, справедлива, если показатель степени в уравнении (1) регрессии $E > 1$. [4, 15, 46].

На практике изредка пользуются упрощенной ***линейной зависимостью трудозатрат*** от размера программного продукта ($E = 1$). Для учета влияния на трудоемкость основных факторов можно использовать коэффициенты изменения трудоемкости (КИТ) производства – **М(i)**, учитывающими влияние ***i*** - ой составляющей совокупных

затрат. В них входят факторы процесса непосредственной разработки, факторы программной и аппаратурной оснащенности, а также квалификация специалистов. При этом затраты на разработку можно представить в зависимости от размера комплекса программ ***П***, корректируемого произведением коэффициентов изменения трудоемкости (КИТ – **M(i)**).

Длительность разработки программных продуктов является важнейшей экономической характеристикой, поскольку часто она определяет общие сроки разработки систем, а значит, быстроту реализации идей в различных областях автоматизации. На основе значений трудоемкости ***С***, размера комплекса ***П*** и выбранных значений **M(i)** могут быть рассчитаны ***длительность Т*** и требуемое среднее ***числа специалистов***:

$$T = G \times C^{H} . \qquad (2)$$

Диапазону размеров современных программных продуктов в три-четыре порядка (до 10 млн. строк) соответствуют приблизительно такой же диапазон изменения трудоемкости и стоимости производства программных продуктов. Однако вариация длительностей производства программных продуктов значительно меньше, чем вариация их трудоемкости, и не превышает десятикратный диапазон от нескольких месяцев до 4 – 5 лет. Длительности разработок ***Т*** ограничены сверху и снизу, и одним из основных факторов, определяющих эти границы, является масштаб комплекса программ – ***П***.

Оценка требуемого среднего числа специалистов для конкретного комплекса программ предварительно может быть рассчитана путем деления величины трудоемкости производства (1) на длительность его производства (2): $N = C/T$. ***Средняя производительность труда*** коллектива специалистов при разработке нового сложного комплекса программ $P = П/С$,конечно, различается для различных классов, размеров и других параметров комплексов программ, однако диапазон этих различий не столь велик как изменения размера, требований к качеству и других факторов.

Модель прогнозирования экономических характеристик проектирования процессов производства программных продуктов COCOMO II

Основными факторами при прогнозировании экономических характеристик производства программных продуктов выше рассматривались, сложность, размер – масштаб комплекса программ и доля повторного использования готовых программных компонентов. Для ***более точного экономического обоснования проектов программных продуктов при детальном проектировании*** целесообразно учитывать влияние требований качества и ряда производственных факторов. Разработка полного содержания функций и структуры программных компонентов, их взаимодействия и интерфейсов, архитектуры всего комплекса программ и базы данных, обычно позволяет до 10% повысить точность определения размера комплекса, и достоверность прогнозирования экономических характеристик до уровня около 5 %.

Для ***прогнозирования трудоемкости проектирования и производства программных продуктов*** (человеко-месяцы) ***в модели COCOMO II*** предложены выражения, уточняющие и детализирующие зависимость (1), представленную выше:

$$C = A \times \Pi^{E} \times \prod_{i=1}^{n} M(i) \quad , \qquad (3)$$

где: $A = 2{,}94; \quad E = B + 0{,}01 \times \sum_{j=1}^{5} F(j); \quad B = 0{,}91.$

В ***детальной модели COCOMO II*** на трудоемкость производства программного продукта учитывается влияние 22 факторов – таблица 1.1. Из них пять – ***масштабные факторы***, характеризуются суммой **F(j)** в значении степени влияния размера комплекса программ ***E***. В уравнении оценки трудозатрат в детальной модели COCOMO II варьируются и применяются факторы масштабирования **F(j)** с целью обобщения воздействия основных параметров при прогнозирования производства программных продуктов (таблицы 1.1 и 1.2). Произведение множителей **M(i)** – 17 факторов, непосредственно влияет на изменение трудоемкости производства в выражении (3).

Таблица 1.1.

Состав факторов детальной модели COCOMO II

Символ	Содержание фактора
	Масштабные факторы
F1	Новизна проекта
F2	Согласованность с требованиями и интерфейсами
F3	Управление рисками и архитектурой проекта
F4	Слаженность работы коллектива
F5	Технологическая зрелость обеспечения разработки
Определяется сумма рейтингов ***F(j)*** для расчетов трудоемкости и длительности	
	Факторы, влияющие на затраты производства
	Требования и характеристики объекта производства
M1	Надежность функционирования
M2	Размер базы данных
M3	Сложность функций и структуры
M4	Требование повторного использования компонентов
M5	Полнота и соответствие документации проекта
	Характеристики коллектива специалистов
M9	Квалификация аналитиков
M10	Квалификация программистов
M11	Стабильность коллектива
M12	Опыт работы по тематике проекта
M13	Опыт работы в инструментальной среде
M14	Опыт работы с языками программирования
	Технологическая среда производства
M15	Уровень инструментальной поддержки проекта
M16	Необходимость распределенной разработки проекта
M17	Ограничения длительности производства продукта
	Аппаратурно-вычислительная среда производства
M6	Ограниченность времени исполнения программ
M7	Ограниченность доступной оперативной памяти
M8	Изменчивость виртуальной среды разработки проекта
Определяется произведение рейтингов ***M(i)*** для расчетов трудоемкости и длительности	

Таблица 1.2.

Масштабные факторы требований к объектам разработки

Факторы	Характеристики факторов					
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий	Сверх высокий
Новизна проекта *F1*	Полностью новый	Во многом новый	Частично новый	В основном известный	Значительно известный	Полностью известный
Согласованность с требованиями *F2*	Строгая согласованность	Допускаются компромиссы	Значительная	Относительная	Незначительная	При неоходимости
Коллективность *F4*	Некоторое взаимодействие	Сложное взаимодействие	Зачастую коллективная работа	Высокая степень взаимодействия	Непрерывное взаимодействие	Активное взаимодействие
Зрелость производства *F5*	CMMI Уровень 1	CMMI Уровень 1	CMMI Уровень 2	CMMI Уровень 3	CMMI Уровень 4	CMMI Уровень 5

При использовании выражений (3) следует выбирать набор факторов (***калибровать модель***), имеющих значения коэффициентов изменения трудоемкости (КИТ) **F(j)** в соответствии с характеристикам ***конкретного проекта***. Для выбора значений **M(i)** в соответствии с ***характеристиками конкретного продукта и среды разработки*** следует использовать таблицы рейтингов трудоемкости для выделенных четырех групп, представленных на – рис. 1.9. Номинальными (средними) в выражении (3) принимаются значения **M (i)** = 1,00, при которых соответствующий фактор не влияет на трудоемкость производства программного продукта. Реально для различных конкретных проектов с учетом их особенностей используется только около половины факторов приведенных в таблице 1.1.

Для прогнозирования длительности производства программных продуктов в модели СОСОМО II более детально рассмотрены исходные данные на основе анализа трудоемкости его производства. При этом рекомендуется учитывать те же наборы факторов, как в

выражении (3), которые применяются при прогнозировании трудоемкости непосредственного производства программного продукта ***C***:

$$T = G \times C^{H}. \qquad (4)$$

где: $G = 3{,}67$;

$$H = D + 0{,}02 \times 0{,}01 \times \sum_{j=1}^{5} F(j) = D + 0{,}02 \times (E - B); \quad D = 0{,}28.$$

Максимальные величины каждого из КИТ производства программных продуктов экспериментально оценены в [38] при предположении, что остальные параметры зафиксированы. В действительности многие факторы взаимно коррелированны. Так, например, высокой сложности комплекса программ обычно сопутствует требование высокой безопасности и надежности функционирования, а также длительная эксплуатация. Ряд факторов отражается одновременно на нескольких КИТ. Воздействие в процессе производства на такие факторы и субъективное стремление специалистов на сокращение определенных видов затрат, в некоторых случаях, может оказываться нерентабельным с позиции снижения полных затрат на производство программного продукта.

Большую роль в повышении экономической эффективности производства программных продуктов играет ***повторное использование готовых программных компонентов*** из других проектов (см. главу 1.3). При этом необходимо учитывать дополнительные затраты для обеспечения возможности повторного использования компонентов вместо разработки компонентов заново каждый раз, и сокращение интегральных затрат на производство продукта при их применении.

При калибровке модели COCOMO II предлагаются следующие последовательные процедуры для конкретного проекта:

- устанавливаются значения и ***сумма рейтингов*** масштабных факторов **F** при которой производится первичное оценивание экономических характеристик;
- выбирается набор, значения и оценивается ***произведение рейтингов*** факторов **M**, оказывающих наибольшее влияние на прогнозируемую трудоемкость производства программного продукта, которые сравниваются с предыдущей оценкой;
- для каждого набора рейтингов выбранных факторов производится расчет и анализ трудоемкости и длительности для конкретного проекта комплекса программ.

Преимущества модели COCOMO II [31, 38] состоят в следую-

щем:

- возможен учет достаточно полной номенклатуры факторов, влияющих на экономические характеристики производства сложных программных продуктов;
- предложены наборы для выбора возможных рейтингов для всей номенклатуры используемых факторов;
- фактические данные подбираются в соответствии с реальными проектами комплексов программ и факторами корректировки, которые могут соответствовать конкретному проекту и предприятию;
- прогнозы производственных процессов являются варьируемыми и повторяемыми;
- метод является достаточно универсальным и может поддерживать различные размеры, режимы и уровни качества продуктов;
- модель хорошо подходит для проектов, между которыми нет существенных отличий относительно размера, сложности или функций процессов производства;
- возможна высокая степень достоверности калибровки с опорой на предыдущий опыт коллектива специалистов;
- результаты прогнозирования сопровождаются обязательной документацией;
- модель относительно проста в освоении и применении.

Недостатки модели COCOMO II:

- все результаты зависят от размера программного продукта – точность оценки размера, оказывает ***определяющее влияние*** на точность прогноза трудозатрат, длительности разработки, численности специалистов и производительности труда разработчиков;
- недостаточно учтено влияние требований функциональной безопасности программного продукта;
- не учитывается зависимость между интегральными затратами и количеством времени, затрачиваемым на каждом этапе проекта;
- не достаточно учитывается внешняя среда производства и применения программного продукта;
- игнорируются многие особенности, связанные с аппаратным обеспечением проекта.

Кроме того, модель COCOMO II представляет ***трудозатраты по сумме всех этапов производства*** (от этапа планирования концепции до этапа поставки программного продукта). Проблемы сопровождения, переноса и повторного использования компонентов трудно описывать в рамках одной и той же модели. При использовании модели COCOMO II предполагается наличие базового уровня трудозатрат, используемого при управлении конфигурацией и обеспечении

качества. В модели не учитываются накладные расходы; расходы на командировки и другие побочные расходы при оценках стоимости. При прогнозировании экономических характеристик производства программных продуктов ***для систем реального времени*** необходимо учитывать затраты на создание моделей внешней среды, которые по величине могут быть сопоставимыми с затратами на основной программный продукт. Системная интеграция и инструментальная поддержка динамического тестирования и испытаний программного продукта реального времени на соответствие требованиям составляют отдельную специфическую задачу [23], которая не входила в состав целей COCOMO II и далее не учитывается.

Влияние масштабных факторов при прогнозировании экономических характеристик производства программных продуктов

Выше описаны, наиболее значительные интегральные факторы модели COCOMO II, влияющие на трудоемкость программных продуктов. Из них ***5 масштабных факторов прогнозирования*** **F (j)** могут существенно изменять трудоемкость и стоимость производства. Остальные 17 факторов **M** объединены в ***четыре группы*** модели COCOMO II (см. рис. 1.9 и таблицу 1.1). Каждый из них может влиять на экономические характеристики производства сложных программных продуктов на уровне 5 – 10%. Разделы данной главы посвящены комментированию факторов из этих групп при этом предполагается, что для основных факторов имеются их описания, представленные в [38]. Рекомендуется выбирать и учитывать те факторы, коэффициенты влияния которых на трудоемкость в конкретном проекте, имеют достаточно большую величину, коррелированную с точностью определения размера комплекса программ или превышают ее. В ***группу*** **F** следует включать совокупность факторов, которые способны значительно изменять трудоемкость:

- новизну проекта комплекса программ;
- необходимую степень согласованности проекта с требованиями технического задания заказчика;
- наличие управления рисками и архитектурой проекта;
- уровень обобщенной квалификации, слаженности и организованности коллектива разработки проекта;
- уровень обеспечения и оснащения технологии производства по оценке зрелости модели CMMI.

Новизну проекта **F1** рекомендуется учитывать при прогнозировании по величине формализации целей и функций программного продукта, наличия у специалистов опыта аналогичных разработок, а также необходимости создания новых архитектуры, компонентов, алгоритмов обработки данных. При этом, в крайних значениях рейтингов предполагается, что полностью отсутствовало проектирование и производство подобного программного продукта или такие продукты были освоены для серийного производства.

Согласованность (жесткость) **F2** проекта программного продукта определена как степень обеспечения соответствия заданным ***требованиям и спецификациям заказчика*** к компонентам, к внутренним и внешним интерфейсам, ко всему комплексу программ в процессе его проектирования и производства. В первом предельном случае считается, что все требования должны быть выполнены обязательно полностью (***заказной продукт***), а в другом крайнем варианте рейтингов эти требования могут рассматриваться только как необязательные пожелания.

В модели COCOMO II значительное внимание уделено ***влиянию организации и взаимодействия коллектива квалифицированных разработчиков*** **F4** на трудоемкость создания сложных программных продуктов (таблица 1.2). В составе организационных характеристик коллектива рекомендуется учитывать согласованность целей специалистов, участвующих в проекте, их психологическую совместимость и способность к дружной коллективной работе, наличие опыта работы в данном коллективе и другие объективные и субъективные свойства участников проекта. При этом большое значение может иметь личная мотивация и психологические особенности поведения специалистов при комплексной работе над проектированием и производством сложного программного продукта.

Эти характеристики обобщены в показатель ***влияния коллективности – сложности взаимодействия специалистов*** **F4** в коллективе, которому сопоставлены рейтинги изменения трудоемкости производства продукта. Наилучшим считается непрерывное корректное взаимодействие организованной ***«команды»*** специалистов с большим опытом работы в данном коллективе при полной согласованности их целей, планов и методов работы. В остальных случаях может отсутствовать глубокое взаимодействие отдельных специалистов, вследствие чего возрастать (даже в несколько раз) трудоемкость производства продукта.

В модели COCOMO II ***для оценивания экономических характеристик при проектировании и производстве*** **F5, *рекомендуется*** учитывать уровень ***методологии*** производства и обеспечения качества сложных программных комплексов на основе ***системы и модели оценки зрелости*** **CMMI** (см. главу 1.1), применяемых технологических процессов. На предприятиях, достигших высокого уровня зрелости, процессы производства должны принимать статус стандарта, фиксироваться в организационных структурах и определять производственную тактику и стратегию корпоративной культуры производства и системы обеспечения качества программных продуктов.

При экономическом прогнозировании проектирования программных комплексов, для достижения устойчивых результатов в процессе развития технологии и организации управления жизненным циклом комплексов программ рекомендуется использовать эволюционный путь, который позволяет совершенствовать, постепенно снижать затраты и повышать качество процессов производства и продуктов. Методология CMMI рекомендует большой комплекс процессов, который предприятие должно выполнять для приобретения, поставки, производства, использования и сопровождения крупных комплексов программ, и виды деятельности, характеризующие степень технологической зрелости этих процессов.

В модели COCOMO II приводятся количественные рекомендации ***коэффициентов влияния уровней зрелости*** **CMMI** ***на трудоемкость*** реализации сложных программных продуктов. Рейтингам технологической зрелости производства в детальной модели COCOMO II сопоставлены уровни CMMI, для каждого из которых приводятся коэффициенты изменения трудоемкости разработки (см. таблицу 1.2). Применение совершенных технологий и инструментальных средств высшего уровня зрелости может снизить трудоемкость производства сложных программных продуктов более чем в 5 раз по сравнению с наиболее широко применяемыми технологиями второго – третьего уровня зрелости CMMI.

Влияние коллектива специалистов при прогнозировании экономических характеристик проектирования и призводства программного продукта

В ***модели COCOMO II*** значительное внимание уделено ***влиянию организации и взаимодействия коллектива специалистов*** на трудоемкость создания сложных программных продуктов (таблица

1.3). В составе организационных характеристик коллектива рекомендуется учитывать: согласованность целей специалистов, участвующих в проекте, их психологическую совместимость и способность к дружной коллективной работе, наличие опыта работы в данном коллективе и другие объективные и субъективные свойства участников проекта. При этом большое значение может иметь личная мотивация и психологические особенности поведения разных специалистов при комплексной работе над производством сложного программного продукта.

Эти характеристики обобщены в качественный показатель ***влияния коллективности – сложности взаимодействия специалистов*** в коллективе, которому сопоставлены рейтинги изменения трудоемкости производства продукта **F4**. Наилучшим считается непрерывное корректное взаимодействие организованных специалистов с большим опытом работы в данном коллективе при полной согласованности их целей, планов и методов работы. В остальных случаях в той или иной степени (даже в 3 – 5 раз) может возрастать трудоемкость производства продукта, что нельзя не учитывать при прогнозировании экономических характеристик и обосновании производства крупных продуктов.

В ***COCOMO II*** – изложены экспертные оценки для учета влияния ***квалификации и характеристик различных категорий специалистов*** на экономические характеристики производства программного продукта [21] (см. таблицу 1.3.). Выбор и использование их значений при прогнозировании экономических характеристик требует высокой квалификации экспертов и детального знания особенностей коллектива специалистов и среды производства реального продукта. Для обеспечения возможности их использования при прогнозировании трудоемкости и длительности разработки сложных комплексов программ в модели COCOMO II представлены соответствующие относительные рейтинги – влияния ряда факторов проектов комплексов программ, и в частности, характеристик специалистов.

Эти данные ***отражают квалификацию специалистов*** по степени их влияния на экономические характеристики производства программных продуктов и тем самым ***уточнять требования к квалификации коллектива специалистов***.

Таблица 1.3.

Характеристики и влияние коллективизма разработчиков программных продуктов на трудоемкость их производства

Коллективизм	Значения характеристик				
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий
Согласованность целей коллектива	Минимальная	Незначительная	Относительная	Значительная	Полная
Способность членов коллектива адаптироваться к целям других	Малая	Незначительная	Относительная	Значительная	Полная
Опыт работы в составе данного коллектива	Нет	Малый	Незначительный	Значительный	Большой
Степень доверия и взаимодействия в коллективе	Нет	В малой степени	В некоторой степени	Значительная	Большая
Обобщенная коллективность работ	Некоторое взаимодействие в коллективе	Сложное взаимодействие	Зачастую коллективная работа	Высокая степень взаимодействия	Непрерывное взаимодействие
Обобщенный коэффициент влияния коллективности *F4*	5,48	4,38	3,29	2,19	1,10

При применении этих рейтингов следует учитывать, что некоторые из них взаимосвязаны, и целесообразно анализировать их корреляцию, возможность объединения или исключения при оценке реальных проектов и коллективов специалистов.

В составе ***организационных характеристик коллектива рекомендуется учитывать факторы,*** представленные в таблице 1.3. При этом большое значение может иметь личная мотивация и ***психологические особенности поведения разных специалистов*** – обобщенная коллективность работ над производ-ством сложного программного продукта. Эти характеристики обобщены в качественный показатель ***влияния коллективности – сложности взаимодействия специалистов*** в коллективе.

Среди ряда ***характеристик коллектива разработчиков***, наибольшее влияние на трудоемкость оказывает ***тематическая и технологическая квалификация специалистов.*** Совместно эти факторы могут изменять трудоемкость производства программных продуктов на 30 – 40%. Кроме того, в модели COCOMO II выделен и обобщен на основе пяти характеристик ***коэффициент – комплексная коллективность участников проекта*** (см. таблицу 1.3), влияние которого может достигать пятикратного изменения трудоемкости.

Организационное разделение специалистов, осуществляющих ***производство программного продукта*** (первая категория), и ***специалистов – технологов***, обеспечивающих, контролирующих и управляющих его качеством в процессе производства (вторая категория), должно обеспечивать эффективное достижение заданных характеристик, а также независимый, достоверный контроль затрат и качества результатов производства.

Специалисты первой категории непосредственно создают компоненты и комплекс программ в целом с заданными показателями качества (см. таблицу 1.4 – факторы **M10, M12, M13, M14**). В процессе производства их функции заключаются в тщательном соблюдении принятой в фирме технологии и в формировании всех предписанных руководствами исходных и отчетных документов.

При этом предполагается, что выбранная технология способна обеспечить необходимые значения конструктивных показателей качества, а достижение заданных функциональных характеристик гарантируется ***тематической квалификацией*** соответствующих специалистов и регулярным контролем этих характеристик в процессе производства.

Таблица 1.4.

Характеристики разработчиков программного продукта

Характеристика	Значение характеристики				
	Очень низкая	Низкая	Номинальная	Высокая	Очень высокая
Квалификация аналитиков *M9*	15%	35%	55%	75%	90%
Квалификация программистов *M10*	15%	35%	55%	75%	90%
Тематический опыт *M12*	2 мес.	6 мес.	1 год	3 года	6 лет
Инструментальный опыт *M13*	2 мес.	6 мес.	1 год	3 года	6 лет
Опыт работы с языками *M14*	2 мес.	6 мес.	1 год	3 года	6 лет
Стабильность коллектива *M11*	48%	24%	12%	6%	3%

Разделение труда специалистов этой категории в крупных коллективах приводит к необходимости их дифференциации ***по профессиональной квалификации и областям деятельности***:

- ***разработчики программных компонентов – программисты*** – **M10** создают компоненты, удовлетворяющие спецификациям требований, реализуют возможности продукта, отслеживают и исправляют ошибки, при производстве сложных программных компонентов, что требует детального знания языков программирования, визуального программирования, сетевых технологий и проектирования баз данных – **M13**, **M14** в таблице 1.4;
- ***тестировщики*** обеспечивают проверку функциональных спецификаций требований, средств обеспечения производительности, пользовательских интерфейсов, разрабатывают стратегию, планы и выполняют тестирование на соответствие требованиям для каждого компонента и комплекса программ;
- ***системные интеграторы*** сложных проблемно-ориентированных комплексов программ работают над продуктом в значительной степени отличными от программистов методами, на разных язы-

ках производства, используют различные средства автоматизации производства и имеют на выходе различные результаты: крупные компоненты и комплексы программ;

- ***специалисты, управляющие сопровождением и конф***игурацией отвечают за снижение затрат на модификацию и сопровождение программного продукта, обеспечение максимальной эффективности взаимодействия компонентов и производство версий программного продукта, принимают участие в обсуждениях интерфейсов и архитектуры продукта;
- ***документаторы*** процессов и объектов жизненного цикла комплекса программ обеспечивают подготовку и издание сводных технологических и эксплуатационных документов на программный продукт в соответствие с требованиями стандартов.

Успех и качество при производстве крупных программных продуктов зависит от слаженности работы и профессионализма коллектива специалистов на всех этапах и уровнях производства продуктов – ***от стабильности коллектива*** – **M11** (табл. 1.1 и 1.4). Особенно важна не индивидуальная характеристика квалификации каждого специалиста, а, прежде всего, ***интегральный показатель квалификации «команды»***, реализующей некоторую, достаточно крупную функциональную задачу или весь программный продукт [8, 51].

Тематическая квалификация и опыт – **M12** ***специалистов*** в конкретной прикладной области, для которой разрабатывается программный продукт, приближенно может оцениваться продолжительностью работы по данной тематике. При низкой тематической квалификации допускаются наиболее грубые системные ошибки, требующие больших затрат при доработке программ. Имеются примеры, когда из-за таких ошибок, допущенных на этапе системного анализа, приходилось в процессе производства изменять 70 – 90% программ. Целесообразность использования в качестве параметра квалификации, значений длительности работы в определенной прикладной области подтверждается достаточно высокой ее корреляцией с коэффициентом изменения трудоемкости. Приводимые в разных работах оценки показывают, что при изменении опыта работы в данной области от 1 до 10 лет производительность труда может повышаться в 1,5 – 2 раза.

Разработчики программного продукта должны иметь в своем составе квалифицированных, ***проблемно-ориентированных аналитиков и системных архитекторов*** – **M9**, способных переводить функ-

циональные требования заказчика в конкретные спецификации и технические требования к комплексу программ и его компонентам (см. табл. 1.1 и 1.4). Уровень квалификации аналитиков в **COCOMO II** предлагается оценивать в процентах от высшей квалификации, что может снизить трудоемкость производства почти на 30% от номинальной.

Технологическая квалификация – **M13** программистов в использовании инструментальной системы автоматизации производства программных компонентов отражает опыт применения методов, средств и всего технологического процесса при создании данного типа программных комплексов. Этот опыт можно характеризовать длительностью работы с конкретной инструментальной системой автоматизации или ее версиями, базирующимися на единых технологических концепциях, опытом и длительностью работы с регламентированными технологиями, инструментальными комплексами автоматизации разработки, ***языками проектирования, программирования и тестирования программ***. Особое значение имеет коллективный опыт организации и выполнения сложных проектов на базе современных автоматизированных технологий и инструментальных средств. Опыт применения конкретного комплекса автоматизации, языков проектирования и программирования может являться существенным фактором при выборе технологии для создания новых компонентов и обеспечении качества программных продуктов.

Оценка производительности коллектива существенно зависит от ***стабильности состава и психологического климата*** (см. **M11** в табл. 1.1 и 1.4) в коллективе специалистов и их способности к сотрудничеству и дружной совместной работе над единым продуктом. В данном факторе годы работы с конкретной технологической системой отражают не только опыт работы с инструментами, но и ***слаженность коллектива*** по проведению больших комплексных работ. Нарушение технологии, задержка при разработке отдельных модулей или групп программ может приводить к большим дополнительным затратам и значительной задержке производства продукта в целом. Однако чаще всего коллективы специалистов имеют уже некоторый технологический опыт, и дальнейшее повышение их квалификации может дать относительно немного 10 – 20% производительности.

Программистская квалификация специалистов – **M10**, **M14** и опыт работы с языками программирования, в наименьшей степени

отражается на производительности труда. В данном факторе учитывается освоенность не только языка непосредственного программирования, но и всех компонентов средств, используемых при создании программ (спецификаций, диалога, отладки, комплексирования). После двух-трех лет работы в наибольшей степени проявляются индивидуальные особенности конкретных специалистов, их творческие способности, тщательность в работе, рациональное использование средств автоматизации. При производстве сложных комплексов программ после первых лет работы возрастание программистской квалификации может повысить производительность труда на 5 – 10%.

Директивное ***ограничение сроков производства продукта*** **M17** относительно типовых – номинальных до уровня 75% от заданных, оптимальных, может сопутствовать значительное снижение качества и увеличение рисков реализации проекта.

Специалисты второй категории – технологи, обслуживающие и сопровождающие технологический инструментарий, который применяется специалистами первой категории, должны обеспечивать применение системы качества предприятия, контролировать и инспектировать процессы производства для минимизации затрат. Основные задачи второй категории специалистов должны быть сосредоточены на контроле ***экономических характеристик*** процессов и результатов выполнения работ, а также на принятии организационных и технологических мер для достижения их необходимого качества, обеспечивающего выполнение всех требований технического задания на программный продукт.

Технологи должны уметь выбирать, приобретать и осваивать экономически наиболее эффективный инструментарий для заказных проектов, реализуемых конкретным предприятием с учетом особенностей создаваемых комплексов программ требуемого качества и рентабельности технологических средств. Они должны разрабатывать регламентированный технологический процесс и систему качества, поддерживающие весь ЖЦ комплексов программ и ***обучать специалистов – разработчиков*** квалифицированному применению соответствующих инструментальных средств и технологий.

Специалисты, управляющие обеспечением качества программных продуктов, должны владеть стандартами и методиками предприятия, поддерживающими регистрацию, контроль, документирование и воздействия на показатели качества и экономические характеристики на всех этапах ЖЦ комплекса программ. Они должны

обеспечивать эксплуатацию системы качества проекта, выявление всех отклонений от заданных показателей качества продуктов и процессов, а также от предписанной технологии на этапах производства.

При ***обобщении рассчитанных экономических характеристик*** некоторые значения могут не удовлетворить заказчика, руководителей и специалистов, ведущих прогнозирование. На практике могут существовать ограничения реальной численности и квалификации коллектива, выделяемого для данного проекта комплекса программ, или допустимой (директивной) длительности производства. Поэтому ***в методике прогнозирования должна допускаться возможность пересчета*** получаемых прогнозных значений с использованием уточненных и откорректированных значений основных факторов и требуемых характеристик специалистов. Согласованные с заказчиком основные экономические характеристики позволяют фиксировать базовый сценарий и проектирование процессов производства продукта, проводить расчеты распределений трудоемкости и длительности по этапам и специалистам.

Влияние технологической и компьютерной среды проектирования и производства при прогнозировании экономических характеристик программных продуктов

Для оценивания влияния на трудоемкость ***инструментальных характеристик производства***, в модели COCOMO II выделен показатель и соответствующий набор рейтингов. ***Инструментальные системы*** – **M15**, поддерживающие производство, могут быть описаны качественными характеристиками и рейтингами, изменяющими трудоемкость в пределах приблизительно 20% от средней – номинальной. Уровень технологии и комплекса инструментальных средств особенно сильно влияет на ***экономику крупных программных продуктов***. Поэтому затраты на их реализацию и применение, целесообразно учитывать конкретно с использованием функций и характеристик проекта. Рейтинги инструментальной поддержки коррелированны с уровнями зрелости технологий производства программ CMMI – **F5** и комментированы выше в главе 1.1.

Затраты на технологию и программные средства автоматизации производства обычно являются весьма весомыми при использовании высокоэффективных автоматизированных технологий и инструментов. При ***экономическом обосновании проекта следует учитывать***, что размер и сложность создаваемого продукта значительно влияет на выбор инструментальных средств и уровня автома-

тизации технологии, а также на долю этих затрат в общих затратах на производство. Возможны ситуации, при которых затраты на технологию достигают 50% общих затрат на производство. Такие затраты могут быть ***оправданы повышением производительности труда, сокращением сроков разработки*** и последующим снижением затрат на множество версий программного продукта.

Стремление уменьшить технологические затраты в период производства без учета последующего использования программного продукта, его компонентов и всего жизненного цикла, может оказаться мало полезным, а в некоторых случаях привести к значительному увеличению совокупных затрат. При применении сложных программных продуктов эти затраты могут исчисляться сотнями человеко-лет, что определяет особую актуальность их снижения. Поэтому необходим системный анализ распределения и использования технологических ресурсов на разработку комплексов программ ***с учетом всего их жизненного цикла***, включая сопровождение и возможный перенос на другие платформы.

Уровень автоматизации, качество технологии и инструментальных средств, используемых для поддержки всего жизненного цикла комплекса программ, обычно сильно коррелирован с достигаемым качеством производства комплекса, а также с качеством средств автоматизации для оценивания этого качества. Оценивание качества технологической базы производства на предприятии позволяет ***прогнозировать возможное качество программного продукта*** и ориентировать заказчика и пользователей при выборе разработчика и поставщика для определенного проекта с требуемыми характеристиками качества производства.

В некоторых комплексах программ реального времени (СРВ) на увеличение трудоемкости разработки (до 50%) может оказать ***ограничение вычислительных ресурсов*** оперативной памяти – **М7** и производительности объектного компьютера – **М6**, на которой должен функционировать программный продукт. При таких ограничениях ресурсов резко усложняется проектирование и производство продукта и приходится адаптировать размеры алгоритмов и программ с целью не нарушить ограничения вычислительных ресурсов. Это привело к необходимости разрабатывать ***методы эффективного использования программами аппаратных ресурсов компьютера***.

На практике наибольшие технические трудности обычно вызывают ***ограничения производительности*** **М6** ***реализующей заказным***

компьютером, для преодоления которых приходится не столько экономить программные ресурсы и сокращать размеры программного продукта, сколько искать пути увеличения производительности компьютера и эффективности их использования. При этом существенным ограничивающим фактором являются реальное время, в течение которого компьютер может быть предоставлен для решения данной задачи, или то время, в пределах которого целесообразно получить результаты (время отклика) для их практического использования. Таким образом, уже на стадии формулирования требований к программному продукту в некоторых проектах (например, бортовые системы) следует сопоставлять доступную производительность компьютера с необходимыми затратами времени для решения требуемого набора задач или с допустимой длительностью ожидания результатов (временем отклика). Поэтому весьма актуальной для систем реального времени является ***проблема эффективного распределения ограниченной производительности компьютера*** для решения ***комплекса заказных функциональных задач*** и поиск методов рациональной организации вычислительного процесса при применении программного продукта.

Быстрый рост количества решаемых задач, их сложности и требуемой производительности вычислительных средств стимулирует поиск различных путей удовлетворения потребностей заказных программных продуктов в ресурсах для решения таких задач. Мировая практика развития вычислительной техники показала, что потребность в вычислительных ресурсах для решения некоторых заказных задач растет быстрее, чем доступные реальные характеристики компьютеров. В таких компьютерах проблема эффективного распределения и использования ресурсов усложняется, так как необходимо оперативно распределять производительность (в принципе различных) вычислительных процессоров для решения всей заданной совокупности задач.

В последние годы стали возможными ситуации, когда ***в современных компьютерах реального времени*** может быть создан такой избыток производительности и памяти, что нет необходимости в детальном выборе и применении методов рационального и экономного использования вычислительных ресурсов при экономическом прогнозировании и обосновании таких проектов. Однако применение более дешевых и простых компьютеров может решать те же задачи при меньших аппаратурных затратах, особенно это актуально для мо-

бильных компьютеров реального времени (например, бортовых систем). При ограничениях ресурсов вследствие требований минимизации весов и габаритов специализированных, реализующих компьютеров в авиационных, ракетных и космических системах, их ***экономное использование остается актуальным*** и следует учитывать при проектировании и производстве соответствующих программных продуктов.

Влияние ***ограничения объема памяти реализующего*** компьютера – **M7** в модели COCOMO II, определяется долей от фактического размера доступной оперативной памяти, которая может использоваться для размещения программного продукта и данных при решении возлагаемых на компьютер совокупности задач с приемлемой для практики точностью и надежностью. Ограничения этого фактора реально меньше влияют на экономические характеристики производства программных продуктов, так как технически легче преодаливаются при конструировании компьютера.

Изменчивость среды разработки – **M8** (технологических средств) может быть вызвана изменением и расширением функций программного продукта, модернизациями устранения ошибок, частота которых вызывает смену версий программного продукта и соответствующие дополнительные затраты на инструментальную среду производства. Номинальной в модели COCOMO II принята стабильность вычислительной среды, когда изменения происходят в среднем каждые 3 месяца. При более частых изменениях среды (каждую неделю) трудоемкость может возрастать на 30%.

В полном жизненном цикле комплексов программ для систем реального времени в этом случае следует прогнозировать ***затраты на компьютеры*** для ***реализации*** программного продукта, автоматизации его разработки – на ***технологическом компьютере*** и для имитации внешней среды – на ***моделирующем*** компьютере. Их применение изменяется по этапам и значительно зависит от требований к качеству продукта, от его сложности, предполагаемого тиража и ряда других факторов. Соотношение между этими затратами зависит от уровня автоматизации производства комплекса программ. По мере возрастания автоматизации повышается доля использование технологических компьютеров, а затем и моделирующих компьютеров, имитирующих внешнюю среду.

Средства автоматизации прогнозирования экономических характеристик производства программных продуктов

На базе модели COCOMO II с коэффициентами из ее таблиц созданы ***программные технологические пакеты программ для прогнозирования*** экономических характеристик проектирования и производства сложных программных продуктов. Модель COCOMO II позволяет определять основные прогнозируемые экономические характеристики, а также уточнять некоторые дополнительные данные для ***экономического обоснования производства конкретного программного продукта*** – рис. 1.10 [24, 38, 46].

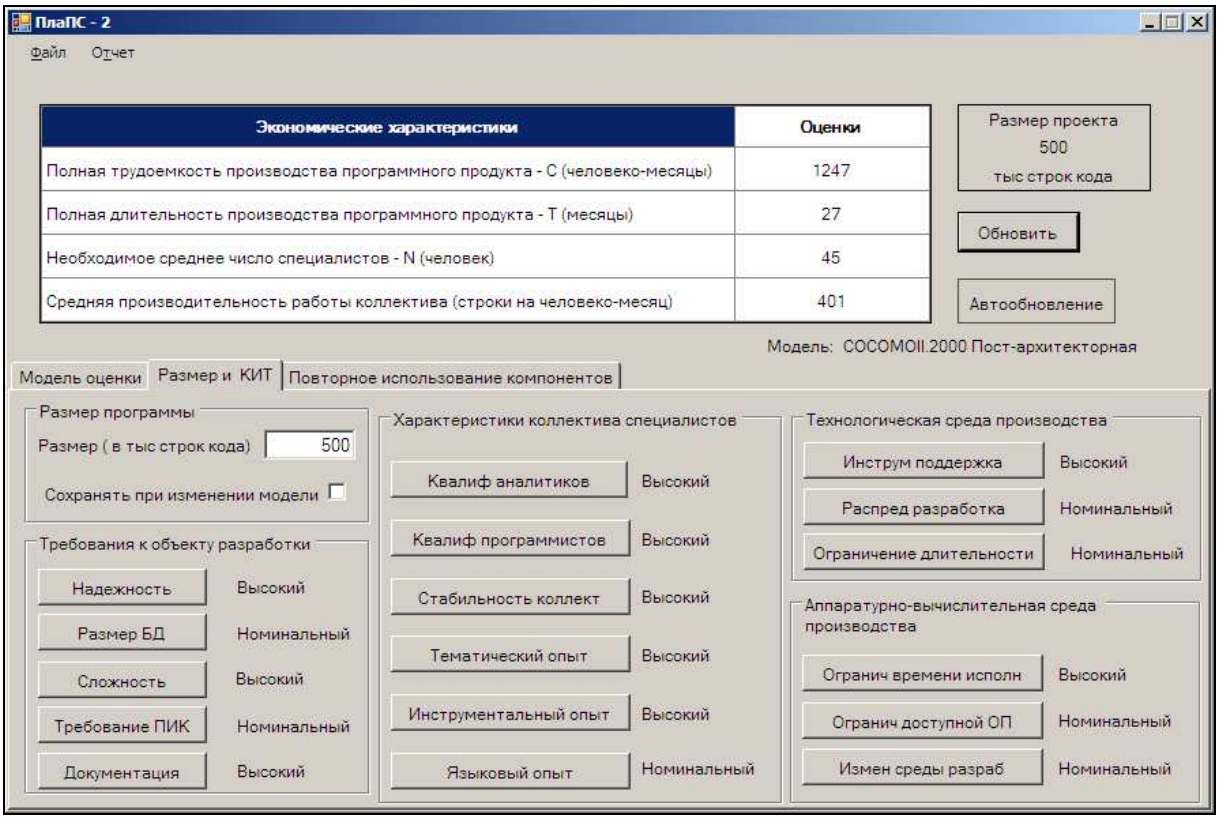

Рис. 1.10

Анализ прогнозирования совокупных затрат и влияния выделенных факторов, позволяет избежать грубых ошибок, связанных с нерациональным планированием распределения ресурсов проекта при ***детальном проектировании и последующем производстве комплекса программ***. При этом обычно значения ряда факторов являются фиксированными в силу объективных условий производства на конкретном предприятии. В отдельных случаях для выбора наилучшей стратегии реализации проекта может быть полезным варьирование нескольких вариантов факторов перед детальным про-

ектированием конкретного комплекса программ. В крайнем случае, на основе выполненных оценок на этапах предварительного или детального проектирования может быть сделано достаточно достоверное ***обоснование возможности прекращения производства*** предполагавшегося программного продукта.

После предварительных оценок экономических характеристик, в процессе производства целесообразно их сравнивать с реальными доступными ресурсами коллектива специалистов и заданными сроками, и при необходимости производить пересчеты с использованием уточненных и откорректированных значений основных факторов. При ***обобщении рассчитанных экономических характеристик*** на практике, некоторые значения, могут не удовлетворить специалистов ведущих прогнозирование. Поэтому в методиках прогнозирования допускается ***возможность пересчета*** получаемых прогнозных значений на основе уточненных значений некоторых факторов.

Например, на базе учтенных при прогнозе факторов, производительность труда может оказаться отличающейся от реальной, характерной для данного предприятия, определявшейся по реализованным предшествующим разработкам. Для того чтобы учесть это различие, предусмотрена возможность ввода реальной производительности специалистов данного предприятия и пересчета всех остальных экономических характеристик. При этом индивидуальная производительность труда отдельных специалистов коллектива может существенно отличаться от «усредненной», заложенной в модель по статистике предыдущих разработок различного размера. Кроме того, на практике могут существовать ограничения реальной численности коллектива, выделяемого для данного проекта комплекса программ, или допустимой (директивной) длительности производства. Для учета таких ограничений должна быть предусмотрена возможность пересчета экономических характеристик при откорректированных значениях некоторых факторов.

Достоверность рассчитанных значений основных экономических характеристик целесообразно проверять путем сопоставления с показателями аналогов, созданных на том же предприятии, наиболее близких к прогнозируемому продукту. Полная стоимость и длительность разработки обычно подлежат согласованию с заказчиком. В процессе согласования уточняются сценарии проектирования и процессы производства, и возможно изменение не только влияния некоторых факторов, но и требований технического зада-

ния на объект и характеристики продукта. Это необходимо, если превышаются допустимые значения стоимости или длительности разработки. Для удостоверения корректности выполненного прогнозирования и исключения грубых ошибок может быть полезным сопоставить полученные результаты на базе модели COCOMO II с оценками по простейшим формулам (1) и (2).

Часть 2

ПРОИЗВОДСТВО ЗАКАЗНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Глава 2.1

ОСНОВНЫЕ ПРОИЗВОДСТВЕННЫЕ ПРОЦЕССЫ СЛОЖНЫХ ЗАКАЗНЫХ КОМПЛЕКСОВ ПРОГРАММ

Стандарты производственных процессов сложных комплексов программ

Процессы жизненного цикла сложных компьютерных систем и программных продуктов регламентируются стандартами **ГОСТ Р ИСО/МЭК 12207.2007, CMMI, ISO 9000** (см. главу 1.1) и комплексом международных и государственных стандартов, представленных в Приложении 1. Стандарты содержат ***основы производственных процессов***, составляющих жизненный цикл сложных систем, охватывают концепции и идеи, определяющие производство систем, вплоть до их снятия с эксплуатации. Они включают процессы заказа и поставки систем, рекомендации для ***экономической оценки*** и совершенствования процессов их жизненного цикла. Процессы в стандартах образуют множество, из которого специалисты предпри-

ятия могут конструировать ***базовые производственные процессы жизненного цикла конкретных систем и программных продуктов***. Их целесообразно селектировать, подбирать и использовать в ***наборах стандартов для производства сложных заказных программных продуктов*** – рис.2.1.

Стандарты технологии производства предназначены для реализации требований к системе, для преобразования требований в полезную продукцию, для создания условий по выпуску продукции, отвечающей ***техническим и экономическим требованиям заказчика***, по использованию продукции и ее функциональных возможностей в соответствии с назначением.

Основные производственные процессы сложных заказных комплексов программ включают:

- стандарты производственных процессов сложных комплексов программ;
  - стандарты ГОСТ Р ИСО/МЭК 12207.2007, CMMI, ISO 9000 и комплекс стандартов, представленных в Приложении 1;
  - базовые производственные стандарты и документы комплексов программ;
- производственные процессы обеспечения качества компонентов и комплексов программ:
  - комплексные системы обеспечения и контроля качества производства программных продуктов;
  - средства и система гарантирования высокого качества программных продуктов;
- производственные процессы верификации и тестирования сложных комплексов программ:
  - верификацию требований и тестов компонентов и комплексов программ;
  - выбор типов и применение эталонов для тестирования компонентов и комплексов программ;
  - связь стратегий последовательностей программирования компонентов, их тестирования и тестовых процедур;
  - удостоверение качества результатов тестирования компонентов и комплексов программ;
- производственные процессы документирования сложных заказных комплексов программ:
  - задачи и особенности технологической и эксплуатационной документации программных продуктов;
  - организация производства, изменения и хранения документации на программные продукты;
- дефекты и ошибки в компонентах и сложных комплексах программ.

Рис. 2.1

Процессы в стандартах помогают выявлять заинтересованных лиц, которые связаны с системой на протяжении жизненного цикла, а также их потребности и пожелания. Технологические процессы производства составляют: определение и анализ требований заказчика, проектирование архитектуры, реализацию и комплексирование системы, а также верификацию, тестирование, аттестацию, документирование систем и программных продуктов. Эти процессы должны

обеспечивать условия для того, чтобы продукция были ***нужной и полезной, экономически выгодной***, пригодной для обслуживания, производства и применения, а также обладала другими конкретными качествами, необходимыми для того, чтобы удовлетворять потребности заказчика, пользователей и поставщика.

Стандарт **ГОСТ Р ИСО/МЭК 12207.2007** – наиболее полно (на уровне международных стандартов) регламентирует ***технологию проектирования, производства и обеспечения качества сложных систем и программных продуктов*** (см. рис. 1.2). Модели и процессы стандарта могут использоваться для представления всего жизненного цикла (ЖЦ) от замысла до прекращения применения или для представления части жизненного цикла, соответствующей текущему состоянию проекта. Жизненный цикл комплексов программ может быть представлен набором этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующих ***экономически эффективное производство*** на всех этапах от подготовки технического задания до завершения испытаний ряда версий и окончания эксплуатации программного продукта.

Стандарт **ГОСТ Р ИСО/МЭК 12207.2007** устанавливает ***строгую связь*** между системой и её программными продуктами (см. рис. 1.2). Она основывается на общих принципах системной инженерии. Программный продукт трактуется как целая ***часть заказной системы***, выполняющий определенные функции в данной системе. Это осуществляется посредством выделения требований к программным комплексам из требований к системе, проектирования, производства и объединения их в систему. Этот принцип является фундаментальной предпосылкой для стандарта, в котором программные продукты всегда существуют в контексте системы. Стандарт может использоваться как непосредственный директивный, руководящий или как рекомендательный документ, а также как организационная база при создании ***экономически эффективных*** процессов и соответствующих технологических этапов производства. Стандарт определяет ***набор проектных и производственных процессов***, ориентированных на большие и сложные программные продукты. Стандарт может быть адаптирован и применен к программному проекту любого типа, размера и сложности.

Основные работы по производству сложного комплекса программ рекомендуется начинать с определения ***состава базовых стандартов и документов производства***, выбора средств конфигу-

рационного управления и обеспечения качества, а также выбора методов и средств технологического обеспечения производства всей системы. Следует проанализировать и формализовать системные требования и критерии качества системы: функциональные, пользовательские, безопасности, интерфейсов с внешней средой, сопровождаемости. На этой базе должна проектироваться архитектура всей системы, выделяться и анализироваться ***требования к программному продукт***у.

Производство системы и программного продукта не должно противоречить политикам, процедурам и нормам организации или национальным законам и регулирующим документам. Каждое такое противоречие должно быть разрешено до начала использования стандартов. Они разработаны для организаций, производящих, приобретающих и сопровождающих системы и программные продукты, а также для поставщиков, разработчиков, операторов, менеджеров (в том числе, менеджеров по качеству) и пользователей программных продуктов.

Рекомендуется разрабатывать план работ, включающий комплексирование компонентов, тестирование по всем разделам требований заказчика и показателям качества, а также документирование плана, результатов интеграции, использованных тестов, критериев оценки и полученных результатов. Далее комплекс программ следует подвергать квалификационному (аттестационному) тестированию по всем разделам требований контракта, при широком варьировании тестов, изменениях значений критериев, а также тестировать полноту и адекватность технологической и эксплуатационной документации реальному программному продукту. Проверенный таким образом программный продукт интегрируется в вычислительные средства системы, средствах визуализации и телекоммуникации. После объединения всех средств, система должна подвергаться квалификационному тестированию и испытаниям на соответствие совокупности требований, а также производится оформление и проверка полного комплекта документации.

Управление производством должно быть сосредоточено, в основном, в подготовке и обеспечении планирования и управления ресурсами, персоналом, аппаратурой, программными средствами и инструментарием. В процессе решения технологических задач должны ***выявляться и регистрироваться проблемы и дефекты*** применения программных продуктов, и их функционирования. Каждые дефект или ошибка должны быть определены, идентифицированы, описаны, проанализированы и устранены в процессе производства и сопровождения в соответствии с контрактом.

Процессы документирования должны охватывать планирование, учитывать рекомендации по стандартизации, проектированию, а также по производству, конфигурационному управлению и сопровождению комплекта документации на комплексы программ. Конфигурационное управление предлагается включать в общий план управления проектом с процедурами конфигурационной идентификации, контроля, учета, отчетности и развития конфигурации.

Производственные процессы обеспечения качества компонентов и комплексов программ

На ***экономические характеристики производства*** большое влияние оказывают требования заказчиков по обеспечению качества, надежности и безопасности применения программных продуктов. Эти требования могут приводить к ***увеличению затрат*** даже в несколько раз. Вследствие роста сфер применения и ответственности функций, выполняемых заказными комплексами программ, резко возросла необходимость ***гарантирования высокого качества программных продуктов***, регламентирования и корректного формирования требований к характеристикам качества и их достоверного определения. Значительные системные и технологические ошибки при определении требуемых показателей качества, оценке трудоемкости, стоимости и длительности производства являются достаточно массовыми и типичными. Многие созданные ***системы не способны выполнять полностью требуемые функциональные задачи с гарантированным качеством*** и их приходится долго и иногда безуспешно дорабатывать для достижения необходимого качества и надежности функционирования, затрачивая дополнительно большие средства и время.

В технических заданиях и реализованных программных продуктах систематически ***умалчиваются и/или недостаточно формализуются понятия и метрики требуемого качества продукта***, каки-

ми характеристиками они описываются, как их следует измерять и сравнивать с требованиями, отраженными в контракте, техническом задании или спецификациях. Кроме того, некоторые из характеристик часто вообще отсутствуют в требованиях и согласованных документах на продукт, что приводит к произвольному их учету или к пропуску при испытаниях [5, 13, 41]. Этому способствует ***ограниченность экономических ресурсов***, особенно времени, необходимых для достижения и оценивания, требуемых значений характеристик качества, а также недостаточная формализация и документирование всего процесса выбора, контроля и анализа качества.

Качество продукта или процесса зависит от того, для какой ***цели***, для какого ***потребителя*** и для каких ***условий*** делается его оценка. Один и тот же объект может иметь несколько различных оценок качества, произведенных для различных целей и разных условий применения. При измерениях и оценках, качество должно рассматриваться как иерархическая совокупность свойств, расположенных на различных уровнях. Каждое из свойств на одном уровне зависит от ряда других свойств, лежащих на более низких уровнях. Число уровней свойств по мере углубления знаний о конкретной продукции или процессе может возрастать. Для получения комплексной оценки может использоваться экспертное определение весомости (приоритета) каждого свойства и в первую очередь должно учитываться влияние этого свойства на ***экономику и эффективность использования данного вида продукции***.

Радикальное повышение качества производства программных продуктов и обеспечение их конкурентоспособности возможно только на базе внедрения современных ***стандартизированных технологий и систем качества***, поддерживающих и контролирующих весь их жизненный цикл. Тщательное специфицирование и оценивание характеристик качества программных продуктов – ключевой фактор обеспечения их адекватного применения. Применительно к программным комплексам ***система обеспечения качества производства*** – это совокупность методов и средств организации управляющих и производственных подразделений предприятия, участвующих в проектировании, производстве и сопровождении комплексов программ с целью придания им свойств, обеспечивающих удовлетворение потребностей заказчиков и потребителей при минимальном или допустимом расходовании ресурсов. Различия фактических и требуемых характеристик качества объектов или процессов должно квалифици-

роваться как дефекты или ошибки и являются первичными стимулами для принятия и реализации решений по изменению определяемых значений качества. Для этого ***необходимы экономические ресурсы***, а также ***воля*** руководителей, организация исполнителей, методы и технология для управления качеством и корректировками программ.

В жизненном цикле сложных комплексов программ для обеспечения их высокого качества целесообразно выделять специалистов, ответственных за ***соблюдением промышленной технологии*** их производства и совершенствования, за измерение и контроль качества комплексов в целом и их компонентов. Для обеспечения качества в конкретных проектах должны быть организованы и стимулированы разработка, освоение и применение современных методов, автоматизированных технологий и инструментальных средств производства, обеспечивающих ***предупреждение или исключение большинства видов дефектов и ошибок*** при производстве и модификации программных комплексов и их компонентов.

Все этапы производства и сопровождения необходимо поддерживать ***методами и средствами*** для измерения качества и определения реальных характеристик программ на любых этапах их ЖЦ, для выявления и устранения дефектов. Наличие достаточно полных эталонов на основе совокупности требований спецификаций заказчика и поэтапная их декомпозиция – необходимая база для измерения реального качества программ. Ограниченность ресурсов при производстве приводит к целесообразности тщательного ***планирования, упорядочения и применения экономичных и эффективных методов автоматизации жизненного цикла*** с целью достижения требуемого качества и достоверного его определения. Для каждого сложного продукта, выполняющего ответственные функции, рекомендуется разрабатывать и применять комплексную систему качества, специальные планы и Программу, методологию и инструментальные средства, обеспечивающие требуемые качество, надежность и безопасность функционирования комплексов программ. Последовательная детализация рекомендаций базовых стандартов должна доводиться до формирования должностных инструкций специалистам, образуя в совокупности иерархический комплекс документов – ***систему качества производства программных продуктов на предприятии***.

Чтобы предприятие могло улучшить качество своей продукции, оно должно иметь проверенный, последовательный и надежный метод для аттестации состояния своих процессов, а также иметь средст-

ва использования ее результатов как часть ***Программы усовершенствования производственных процессов***. Использование аттестации производственных процессов внутри предприятия должно способствовать выработке ***культуры постоянного совершенствования*** и повышения характеристик качества программных продуктов, а также соответствующих механизмов поддержания этой культуры и оптимизации использования ресурсов. Это должно приводить к появлению ***зрелых предприятий***, максимально восприимчивых к возрастающим требованиям заказчиков, потребителей и рынка, обеспечивающих ***высокие экономические характеристики***, минимальную стоимость полного жизненного цикла своей продукции и, как результат, максимально удовлетворяющих требования заказчика и пользователей к качеству продуктов [29].

Покупателям и заказчикам программных продуктов выгодно использование аттестации процессов ЖЦ при ***определении производственной зрелости поставщика***, что:

- уменьшит неопределенность при выборе поставщиков программных комплексов за счет того, что риски, связанные с реальной зрелостью подрядчика, выявляются еще до заключения договора;
- позволит заранее предусмотреть необходимые меры на случай возникновения рискового события;
- предоставит количественные критерии выбора при сопоставлении потребностей бизнеса, ***требований и оценочной стоимости производства продукта*** с реальной зрелостью конкурирующих поставщиков;
- приведет к общему пониманию необходимости использования результатов аттестации для усовершенствования процессов, оценки ***экономической и технологической зрелости*** поставщика при прогнозировании характеристик производства комплексов программ. Это помогает определить, ***эффективны ли экономически производственные процессы*** для достижения заданных целей заказного программного продукта, а также выявить существенные причины недостаточного качества продукции, ***риски превышения бюджет или сроков***

Производственные процессы верификации и тестирования сложных комплексов программ

Для обеспечения качества программного продукта необходима верификация и ***трассирование требований*** к функциям и компонен-

там комплекса, а также обеспечение баланса требований к ним. ***Цели верификации*** состоят в том, чтобы обнаружить и зарегистрировать ошибки, которые могли быть внесены в комплекс программ и компоненты во время его производства (см. главу 2.2) [2, 16, 33]. ***Результаты верификации комплекса программ*** должны быть достигнуты посредством выполнения комбинации из просмотров, анализов, разработки тестовых сценариев и процедур и последующего выполнения этих тестовых процедур. Просмотры и анализы должны обеспечивать оценку точности, полноты и верифицируемости требований, архитектуры комплекса программ, компонентов и исходного кода. Разработка сценариев верификации должна обеспечивать оценку внутренней непротиворечивости и полноты состава требований к компонентам, а также демонстрацию соответствия продукта требованиям.

Для обеспечения высокого качества программного продукта параллельно с ***верификацией требований*** и разработкой корректировок, следует разрабатывать и верифицировать ***спецификации и сценарии тестов***, отражающие методы и конкретные процедуры проверки реализации этих требований. Тестовые спецификации могут использоваться для проверки согласованности, внутренней непротиворечивости и полноты реализации требований без исполнения программ. Для каждого требования к функциям комплекса программ, его архитектуре, функциональным компонентам должна быть разработана спецификация требований к тестам, обеспечивающая проверку корректности, адекватности и возможности в последующем реализовать тестирование компонента на соответствие этому требованию. Такая взаимная проверка функций компонентов, отраженных в требованиях и в спецификациях тестов, обеспечивает повышение их качества, сокращение дефектов, ошибок, неоднозначностей и противоречий.

Независимая разработка спецификаций тестов на основе спецификаций требований, создает базу для обнаружения, какие требования не тестировались или принципиально не могут быть проверены тестированием. Таким образом, ***верификация спецификаций требований тестов*** к функциям и характеристикам программных компонентов и комплекса могут использоваться с ***двумя целями***:

- для проверки программ и интерфейсов взаимодействия программных компонентов разных уровней в комплексе программ;
- для создания требований к скоординированному комплексу тестов для проверки совокупности компонентов, обеспечивающих

взаимную проверку реализации спецификаций требований комплексом программ.

Тестирование комплексов программ может иметь ***две цели***, которые одинаково важны для качества программного продукта. Первая цель – тестирование для идентификации дефектов подразумевает успешность процедуры тестирования, если ***дефект найден и устранен***. Это отличается от подхода в тестировании (вторая цель), когда тесты исполняются для демонстрации того, что программный комплекс удовлетворяет предъявляемым требованиями и, соответственно, тест считается успешным, если ***не найдено дефектов***. Тестирование программ может использоваться для демонстрации наличия дефектов, но никогда не гарантирует их отсутствие. Основная причина этого в том, что полное, всеобъемлющее тестирование недостижимо для реального сложного программного продукта.

Обязанностью администраторов программного проекта является ***управление связями между производственными процессами и тестированием***:

- полный план тестирования должен быть определен в начале жизненного цикла программного комплекса;
- должны быть установлены правила регистрации, хранения, модернизации и сопровождения комплекса программ, тестовых данных и среды тестирования;
- необходим интегрированный план сборки системы и компонентов программного комплекса в соответствии со стратегией выпуска версий системы и/или программного продукта;
- должно быть представлено подтверждение функциональных возможностей и качества программного продукта, показывающее их соответствие текущим требованиям заказчика;
- для обеспечения экономически эффективного процесса управления производством следует учитывать ресурсы необходимые для генерации тестов, используемые в процессе тестирования, и их результатов.

Программные продукты обычно ***встраиваются*** в различные сложные ***заказные системы реального време***ни. Работа над такими продуктами требует от программистов и тестировщиков, освоения ***общих проблем проектирования, производства и применения заказных систем*** определенного назначения и сферы использования. Программному специалисту необходимо участвовать в разработке требований для всей системы, а также освоить прикладную область

применения создаваемого комплекса программ до начала тестирования функций, их характеристик и тестов, требованиям которых должен отвечать программный продукт [3, 9, 35]. В жизненном цикле крупных заказных систем и комплексов программ, определяющими их назначение, создание, свойства и качество являются ***три ключевых объекта***:

- ***требования*** заказчика и/или пользователей к системе, которые определяют ***цели создания и назначение программного продукта, его функции, качество и область применения***, необходимые пользователям системы и иным заинтересованным лицам, в заданной эксплуатационной среде;
- ***тесты – эталоны и вторая адекватная форма описания содержания, функционирования и качества программного продукта*** для проверки полноты реализации требований, а также для верификации соответствия исходным требованиям к системе, комплексу и компонентам программ;
- ***эксплуатационная документация***, которая обеспечивает применение программного продукта пользователями в соответствии с требованиями к его функциями и характеристика, и является ***третьем адекватным эталоном*** корректности их реализации.

В ***требованиях к программному продукту (первый эталон)*** должно быть зафиксировано соглашение между заказчиком и выполняющими проект специалистами, отражающее потребности заказчика и пользователей в таком виде, чтобы разработчик мог построить удовлетворяющий их комплекс программ и его компоненты [9, 13, 20]. Требования являются ***исходным эталоном*** при разработке любой системы или программы, должны быть достаточно конкретными, чтобы можно было при тестировании и испытаниях установить, когда они удовлетворены. Вся разработка проекта должна вытекать из требований, а все спецификации программного комплекса и компонентов найти отражение в процессах, функциях и качестве продуктов производства. Эти данные должны устанавливать состав, содержание и значения результатов исполнения программ, которые следует получать системе или пользователям при определенных условиях и исходных данных.

При разработке требований к программным продуктам необходимо учитывать функции, основные свойства и общие требования к проектам систем. Формирование назначения, функций и технического задания на проект системы должно включать основные тре-

бования к комплексу программ и особенности требований к нему заинтересованных лиц. Общие требования к качеству функционирования программных продуктов реального времени обязательно включают: требования к надежности и функциональной безопасности продуктов; требования к производительности и эффективности использования ресурсов компьютеров программными продуктами в реальном времени, а также к допустимым рискам при их применении.

В требованиях должно устанавливаться ***ожидаемое поведение системы и программного продукта***, выраженное, желательно, количественными характеристиками, а также границы их допустимых значений. Требования могут зависеть от абстрактных представлений о подходящих характеристиках системы и включать методы и виды моделирования функционирования и среды, для достаточно полного описания заданных требований. Требуется удостоверять, что требования являются, с одной стороны, ***необходимыми и достаточными*** для удовлетворения заказчика, а с другой – необходимыми и достаточными входными данными для других процессов, в частности для проектирования функций, качества и архитектуры комплекса программ.

При формировании требований следует учитывать, что ***источниками ошибок*** в комплексах программ являются специалисты – конкретные люди с их индивидуальными особенностями, квалификацией, талантом и опытом (таблица 2.1). При этом можно выделить некоторые предсказуемые дефекты требований, модификаций, расширения и совершенствования комплекса, и необходимые изменения, обусловленные выявлением случайных, непредсказуемых дефектов и ошибок [23, 33]. Вследствие этого плотность потоков и размеры необходимых ***корректировок в требованиях*** к комплексу и компонентам программ могут различаться в десяток раз.

Однако в сложных комплексах программ статистика и распределение типов ошибок и выполняемых изменений для коллективов разных специалистов нивелируются и проявляются достаточно общие закономерности, которые могут использоваться как ориентиры при их выявлении. Каждому типу необходимых корректировок соответствует более или менее определенная категория специалистов, являющихся источником дефектов данного типа. Такую корреляцию целесообразно учитывать как ***общую качественную тенденцию*** при анализе и поиске их причин.

Таблица 2.1.

Специалисты – источники и типы первичных дефектов и ошибок

Специалисты – источники дефектов и ошибок	Типы первичных дефектов и ошибок программных комплексов и документации
Заказчик проекта программного продукта	Дефекты организации проекта и исходных требований заказчика на программный продукт
Менеджер проекта программного продукта	Дефекты, обусловленные реальной сложностью программного продукта
Менеджер-архитектор комплекса программ	Ошибки организации проектирования и архитектуры комплекса программ
Проблемно-ориентированные аналитики и системные архитекторы	Системные и алгоритмические дефекты и ошибки комплекса программ
Спецификаторы компонентов комплекса программ	Алгоритмические ошибки компонентов и документов комплекса программ
Системные интеграторы комплекса программ	Системные ошибки и дефекты реализации версий программного продукта и документации
Тестировщики требований и реализации компонентов и комплекса программ	Пропущенные системные, программные и алгоритмические ошибки компонентов, комплекса программ и документации
Управляющие сопровождением и конфигурацией версий программного продукта	Ошибки проектирования и реализации версий программного продукта
Документаторы программного продукта	Дефекты и ошибки технологических и эксплуатационных документов программного продукта

Одной из основных причин ошибок в комплексах программ являются ***организационные дефекты*** при определении и использовании требований к качеству программного продукта, которые отличаются от остальных типов и условно могут быть выделены как самостоятельные. Ошибки и дефекты данного типа появляются из-за недостаточного понимания коллективом специалистов целей и функций комплекса программ, а также вследствие отсутствия четкой его организации и поэтапного контроля требований качества продуктов. Это порождается ***пренебрежением руководителей к организации технологического процесса*** формализации требований сложных программных продуктов и приводит к серьезной недооценке их дефектов, а также трудоемкости и сложности их выявления и устранения.

Реализация этих целей верификации и тестирования может производиться разными методами и специалистами – программистами, интеграторами и тестировщиками, что позволяет использовать результаты их деятельности для сравнения одних и тех же описаний программ, представленных на языках программирования и на языках тестов. Это позволяет выявлять вторичные дефекты, и повышать качество путем ***сопоставления двух методов*** и результатов описания ***одних и тех же*** функций и характеристик программ, за счет того, что мала вероятность одинаковых ошибок в сценариях и реализациях тестов и в описаниях требований к функциям и характеристикам программ.

Для обеспечения эффективности затрат ресурсов тестирование должно быть интегрировано как можно раньше с основными процессами проектирования, разработки и сопровождения в жизненном цикле программных комплексов. По существу, требования к комплексу программ и тесты для проверки их адекватности и полноты, ***отражают один и тот же объект***, но в разной форме. Поэтому сложность представительного описания комплекса тестов ***соизмерима*** со сложностью описания полной совокупности требований к функциям, характеристикам и качеству соответствующего комплекса программ.

Создание плана тестирования – итеративный процесс, требующий обратной связи с различными участниками проекта и согласованности с определенными в нем процессами, стратегиями тестирования и сроками выполнения работ. С этим планом должен быть корелирован и предшествовать план программирования и подготовки к тестированию модулей и компонентов сложных функциональных за-

дач. Тест-менеджер должен ***утвердить стратегию программирования компонентов, их тестирования*** и тестовые процедуры, которые должны быть подробно описаны в плане тестирования, и определять какие компоненты и модули, сценарии и тесты когда будут выполняться. Подробное изучение системных требований или сценариев применения системы вместе с тщательным определением параметров плана тестирования и требований к тестам необходимы для эффективного тестирования программного комплекса.

Тестирование всегда предполагает компромисс между ограниченными экономическими ресурсами и заданными сроками, с одной стороны, и практически неограниченными требованиями к качеству результатов тестирования и программного продукта, с другой. Оценка стоимости и затрат, а также другие измерения процессов, связанных с оценкой ресурсов, необходимых для тестирования, как и оценка ***экономической эффективности тестирования*** на разных этапах и уровнях, основывается на точке зрения и практиках менеджеров проектов и используется для оценки и улучшения процесса тестирования. Учет соответствия между стоимостью или затратами, необходимыми для того или иного метода, и эффективностью тестирования является обязательной частью эффективного управления проектами сложных программных комплексов.

Характеристики качества заказных встроенных программных продуктов реального времени зависят не только от их внутренних свойств, но и ***от свойств внешней среды – сценариев и генераторов динамических тестов***, в которой они испытываются и применяются. Для сокращения неопределенностей и прямых ошибок при оценивании качества комплексов программ необходимо до начала разработки определять основные параметры внешней среды, при которых должен функционировать комплекс программ с требуемыми характеристиками. Для этого заказчик и разработчики совместно должны структурировать, описать и ***согласовать модель внешней среды*** и ее параметры в среднем, типовом режиме применения программного продукта, а также в наиболее вероятных и критических режимах генерации тестов, в которых должны обеспечиваться требуемые характеристики качества функционирования. Увеличение необходимости проверять широкий набор требований и характеристик сложных программных комплексов приводит к необходимости ***динамического генерирования тестов*** посредством моделей внешней среды, охватывающих по возможности весь спектр данных, воздействующих на

тестируемый объект. Подготовку тестов для испытаний крупных комплексов программ целесообразно осуществлять с применением автоматизированной генерации динамических тестов внешней среды в реальном времени, а также средств обработки результатов динамического тестирования.

Производственные процессы документирования сложных комплексов программ

Тексты и объектный код комплексов программ ***могут стать программным продуктом*** только ***в совокупности с документами***, полностью соответствующими их содержанию и достаточными для его освоения, применения и изменения. Для этого документы должны быть корректными, строго адекватными текстам программ и содержанию баз данных – систематически, структурировано и понятно изложены, для возможности их успешного освоения и использования достаточно квалифицированными специалистами различных рангов и назначения. Качество и полнота отображения в документах процессов и продуктов в жизненном цикле программных комплексов, должны полностью определять достоверность информации для взаимодействия заказчиков, пользователей и разработчиков, а тем самым, корректность функций, ***качество программных продуктов*** и соответствующих систем.

Создание программного продукта требует значительных интеллектуальных и организационных усилий, а его документация – это сложный компонент, подверженный изменениям, которые могут вноситься многими специалистами. Управление документацией должно непрерывно поддерживать ее полноту, корректность и согласованность с содержанием программного продукта, что требует значительных затрат. Необходимо обеспечивать возможность достоверного, формально точного общения всех участников производства между собой, с создаваемым продуктом и с документами для гарантии поступательного развития и совершенствования комплекса программ.

Адекватность документации требованиям, состоянию текстов и объектных кодов программ должна инспектироваться и удостоверяться (подписываться) ответственными руководителями и заказчиками проекта. ***Ошибки и дефекты документов не менее опасны для применения и изменения программного продукта***, чем ошибки в структуре, интерфейсах, файлах текстов программ и в содержании данных. Поэтому к разработке, полноте, корректности и качеству документа-

ции необходимо столь же тщательное отношение, как к производству и изменениям текстов программ и данных [5, 20, 33].

Реализация корректных документов в значительной степени определяет достигаемое качество сложных программных продуктов, трудоемкость и длительность их создания. Для этого должна формироваться и использоваться регламентированная стратегия, стандарты, распределение ресурсов и планы создания, изменения и применения документов на программы и данные сложных систем. Официальная, описанная и утвержденная стратегия документирования должна устанавливать ***дисциплину***, необходимую для эффективного создания высококачественных документов на продукты и процессы в жизненном цикле комплексов программ.

Обычно целесообразно выделять специалистов, непосредственно ответственных за создание, адекватность и контроль полноценного комплекта документов на программный продукт. Состав и общий объем документов широко варьируется в зависимости от класса и характеристик объекта разработки, а также в зависимости от используемой технологии. Наиболее сложному случаю разработки критических программных продуктов реального времени высокого качества соответствует самая широкая номенклатура документов**.** Такой перечень документов может быть использован как базовый для формирования на его основе состава и шаблонов документов в остальных более простых проектах [24].

Технологическая документация должна ***отражать производственные процессы*** комплексов программ и данных, требования к ним, включать подробные технические описания, подготавливаемые для специалистов, ведущих проектирование, разработку и сопровождение комплексов программ, обеспечивающие возможность их отчуждения, детального освоения, развития и корректировки программ и данных. Стандарты и нормативные документы должны регламентировать структуру, состав производственных этапов, работ и подготовки документов. Они должны: формализовать выполнение и документирование конкретных работ при проектировании, разработке и сопровождении комплексов программ; обеспечивать адаптацию документов к характеристикам среды разработки, внешней и операционной системы; регламентировать процессы обеспечения качества комплексов и его компонентов, методы и средства их достижения, реальные ***значения достигнутых показателей качества***. Для контроля возможных изменений целесообразно предусматривать и согласовывать с заказ-

чиком специальный документ, регламентирующий правила применения и корректировки их номенклатуры, а также состава и содержания документации, поддерживающей производство комплексов программ.

Эксплуатационная документация должна обеспечивать отчуждаемость программного продукта от их первичных поставщиков – разработчиков, возможность освоения и эффективного применения комплексов программ достаточно квалифицированными специалистами – пользователями для решения конкретных функциональных задач систем. Эксплуатационные документы должны исключать возможность некорректного использования программных продуктов за пределами условий эксплуатации, при которых документами ***гарантируются требуемые показатели качества*** их функционирования. Основная их задача состоит в фиксировании, полноценном использовании и обобщении результатов функционировании программных продуктов и системы.

Контроль адекватности и качества документов является органической функцией управления проектом и должен иметь средства регулирования поведения отдельных специалистов и коллектива разработчиков в целом. В плане управления документированием и обеспечением качества внимание специалистов должно акцентироваться на анализе достигнутых результатов разработки, методах и средствах достижения заданных заказчиком характеристик комплекса программ. При планировании и разработке комплекс документации должен проверяться и аттестоваться на выполнимость и полноту в условиях ограниченных ресурсов, а также на корректность, адекватность и непротиворечивость отдельных документов.

Различия в организационных службах предприятий, процедурах, методах и стратегиях создания программных продуктов, масштабах и сложности проектов, требованиях систем и методах их разработки влияют на ***способы разработки, применения и сопровождения документов***. Сложность, количество и полнота содержания комплекса документов,в первую очередь,зависят от ***масштаба – размера программного продукта***, что целесообразно оценивать в начале его проектирования. Состав и шаблоны документов целесообразно базировать на серии международных стандартов (см. Приложение 1). Менеджеры проекта программного продукта, ответственны за ***выбор экономически эффективной модели комплекса документации*** для конкретного проекта и за адаптацию производственных процессов и шаб-

лонов документов применительно к выбранной модели документооборота проекта программного средства.

Для хранения, тиражирования и распространения документов сложных комплексов программ высокого качества целесообразно выделять группу специалистов, ответственных за ***контроль, обеспечение и гарантированное сохранение документации***. Для критических, важных систем документация на программы и данные должна храниться и дублироваться на различных типах носителей и эпизодически выводиться на бумажные носители. Для обеспечения достоверных данных об объектах и процессах управления документами программных продуктов необходима автоматизированная ***база данных – информационная система обеспечения и хранения документов проектов***.

Дефекты и ошибки в компонентах и сложных комплексах программ

Для эффективной организации процессов производства руководителям и специалистам ***необходимо знать и учитывать основные источники и типы дефектов и ошибок***, возможные в сложных комплексах программ, которые следует изучать, прогнозировать и устранять при производстве. ***Понятие дефекта или ошибки в программе*** подразумевает неправильность, погрешность или неумышленное искажение объекта или процесса, что может быть ***причиной ущерба – риска*** при функционировании и применении программного продукта. При этом должно быть ***известно или задано требование или правильное, эталонное состояние объекта или процесса***, по отношению к которому может быть определено наличие отклонения – ошибка или дефект. Исходным эталоном обычно является спецификация требований заказчика или потенциального пользователя, предъявляемая к программному компоненту или комплексу. Любое отклонение результатов функционирования программы от предъявляемых к ней требований и сформированных по ним эталонов – тестов, следует квалифицировать как ***ошибку – дефект в программе***, наносящий некоторый ущерб при ее применении. Различие между ожидаемыми и полученными результатами функционирования комплекса программ могут быть следствием ошибок не только в созданных программах, но и ошибок в первичных требованиях спецификаций, явившихся базой при создании эталонов. Тем самым проявляется объективная реальность, заключающаяся в невозможности абсолютной корректности и полноты исходных требований и эталонов для программных компонентов и комплексов.

Источниками ошибок в комплексах программ являются специалисты – конкретные люди с их индивидуальными особенностями, квалификацией, талантом и опытом. При этом можно выделить де-

фекты требований, расширения или совершенствования компонентов и комплексов программ, и необходимые изменения, обусловливающие выявление ***случайных, непредсказуемых дефектов и ошибок***. Однако в сложных комплексах программ статистика и распределение типов ошибок и выполняемых изменений для коллективов разных специалистов нивелируются и проявляются достаточно общие закономерности, которые могут ***использоваться как ориентиры*** при их выявлении. Каждому типу необходимых корректировок соответствует более или менее определенная категория специалистов, являющихся источником дефектов данного типа (таблица 2.1). Такую корреляцию целесообразно учитывать как ***общую качественную тенденцию*** при анализе и поиске причин ошибок.

Статистика ошибок и дефектов в компонентах и комплексах программ, и их характеристики в сложных заказных продуктах, могут служить ***ориентирами*** для разработчиков при распределении ресурсов в жизненном цикле комплексов программ и предохранять их от излишнего оптимизма при оценке достигнутого качества программных продуктов. ***Изучение и прогнозирование*** характеристик дефектов и ошибок в программах непосредственно связано с достигаемой корректностью, безопасностью и надежностью функционирования комплексов программ и ***помогает***:

- оценивать реальное состояние проекта и планировать необходимую трудоемкость и длительность для его завершения и устранения доступных ошибок;
- выбирать методы и средства автоматизации тестирования компонентов комплекса программ, адекватные текущему состоянию производства, наиболее эффективные для устранения определенных видов дефектов;
- рассчитывать необходимую эффективность контрмер и дополнительных средств оперативной защиты от потенциальных дефектов и не выявленных ошибок;
- оценивать требующиеся ресурсы, при модификации программ для устранения дефектов и ошибок, с учетом затрат на реализацию контрмер.

На практике исходные требования – эталоны поэтапно уточняются, модифицируются, расширяются и детализируются по согласованию между заказчиком и разработчиками. Базой таких уточнений являются ***неформализованные представления и знания*** специалистов – заказчиков и разработчиков, а также результаты промежуточных этапов проектирования и тестирования. Однако установить не корректность таких эталонов еще труднее, чем обнаружить дефекты в программах, так как принципиально отсутствуют точные, формали-

зованные данные, которые можно использовать как исходные эталоны. В процессе декомпозиции и верификации исходной спецификации требований, возможно появление ошибок в спецификациях на компоненты программ и на отдельные модули. Это способствует расширению спектра возможных дефектов и вызывает необходимость создания ***гаммы методов и средств тестирования*** для выявления некорректностей в спецификациях на компоненты разных уровней.

Важной особенностью процесса выявления ошибок в программах обычно является ***отсутствие полностью определенной программы-эталона***, которой должны соответствовать текст и результаты функционирования разрабатываемой программы. Поэтому установить наличие и локализовать дефект непосредственным сравнением с программой без ошибок в большинстве случаев невозможно. При тестировании обычно сначала обнаруживаются ***вторичные*** ошибки и ***риски***, т.е. последствия и результаты проявления некоторых внутренних дефектов или некорректностей программ. Эти внутренние ***дефекты*** следует квалифицировать как ***первичные*** ошибки или причины обнаруженных аномалий результатов. Последующая локализация и корректировка таких первичных ошибок должна приводить к устранению ошибок, первоначально обнаруживаемых в результатах функционирования программ.

Потери эффективности и риски программ за счет неполной корректности в первом приближении можно считать пропорциональными вторичным ошибкам в выходных результатах. Типичным является случай, когда одинаковые по величине и виду вторичные ошибки в различных результирующих данных существенно различаются по своему воздействию на общую эффективность и риски применения комплекса программ. Таким образом, оценка последствий, отражающихся на вторичных ошибках и функционировании программ, может, в принципе, производиться ***по значениям ущерба – риска вследствие не устраненных их причин – первичных ошибок в программе***. Вторичные ошибки являются определяющими для эффективности функционирования программ, однако не каждая первичная ошибка вносит заметный вклад в выходные результаты. Вследствие этого ряд первичных ошибок может оставаться не обнаруженным и, по существу, не влияет на функциональные характеристики компонента или комплекса программ.

При производстве наибольшее ***число первичных ошибок*** вносится на этапах системного анализа, программирования, разработки

или модификаций текстов программ. При этом на долю системного анализа приходятся наиболее сложные для обнаружения и устранения дефекты. Общие тенденции состоят в быстром росте затрат на выполнение каждого устранения ошибки на последовательных этапах процессов разработки компонентов и комплекса программ. При системном анализе интенсивность обнаружения ошибок относительно не велика, и ее трудно выделить из процесса проектирования компонента или комплекса программ. Интенсивность проявления и обнаружения вторичных ошибок наиболее велика на этапе активного тестирования и автономной отладки программных компонентов. Различия интенсивностей ***устранения первичных ошибок, на основе их вторичных проявлений***, и внесения первичных ошибок при корректировках программ определяют скорость достижения заданного качества компонентов и комплексов программ.

Практический опыт показал, что ***наиболее существенными факторами, влияющими на характеристики обнаруживаемых ошибок, являются***:

- методология, технология и уровень автоматизации системного и структурного проектирования компонентов и комплекса программ, а также непосредственного программирования компонентов;
- длительность с начала процесса тестирования компонентов и комплекса, и текущий этап производства или сопровождения комплекса программ;
- класс комплекса программ, масштаб (размер) и типы компонентов, в которых обнаруживаются ошибки;
- методы, виды и уровень автоматизации верификации и тестирования, их адекватность характеристикам компонентов и потенциально возможным в программах ошибкам;
- виды и достоверность эталонов – тестов, которые используются для обнаружения ошибок.

Одной из ***основных причин ошибок*** в сложных комплексах программ являются ***организационные дефекты создания требований и эталонов к программному продукту***, которые отличаются от остальных типов. Ошибки и дефекты этого типа появляются из-за недостаточного понимания руководителями и коллективом специалистов ***целей и функций комплекса программ***, а также вследствие отсутствия четкой их организации и поэтапного контроля требований качества компонентов и продуктов. Это порождается пренебрежением руководителей к организации всего технологического процесса

формализации требований сложных программных продуктов и приводит к серьезной недооценке их дефектов, а также к трудоемкости и сложности их выявления. При отсутствии при производстве планомерной и методичной разработки и тестирования требований и эталонов может оставаться не выявленным значительное количество ошибок, и, прежде всего, дефекты требований к взаимодействию отдельных функциональных компонентов между собой и с внешней средой. Для сокращения этого типа массовых ошибок, активную роль должны играть лидеры – менеджеры и аналитики – системотехники, способные вести контроль и конфигурационное управление требованиями, изменениями и развитием версий и компонентов комплексов программ.

Системные ошибки и недостатки определения требований к программному продукту характеризуются, прежде всего, неполной информацией о реальных процессах, происходящих в источниках и потребителях информации. Кроме того, эти процессы зачастую зависят от самих алгоритмов и поэтому не могут быть достаточно определены и описаны заранее без исследования функционирования комплекса программ во взаимодействии с внешней средой. На начальных этапах разработки не всегда удается точно и полно сформулировать целевую задачу всей системы, а также целевые задачи основных функциональных групп программ, и эти задачи уточняются в процессе проектирования. Во многих случаях отсутствует полная адекватность условий получения предполагаемых и реальных ***характеристик внешней среды***, что может являться причиной сложных и трудно обнаруживаемых системных ошибок.

Ошибки определения характеристик системы и внешней среды, принятых в процессе производства комплекса программ за исходные, могут быть результатом аналитических расчетов, моделирования или исследования аналогичных систем. В ряде случаев может отсутствовать полная адекватность предполагаемых и реальных характеристик, что является ***причиной сложных и трудно обнаруживаемых системных ошибок и дефектов развития проекта***.

Сложность обнаружения и устранения ошибок значительно конкретизируются и становятся измеримой, когда устанавливается связь этого понятия с конкретными ресурсами, необходимыми для решения соответствующей задачи и возможными проявлениями дефектов. При разработке и сопровождении программ основным лимитирующим ресурсом обычно являются допустимые трудозатраты

специалистов, а также ограничения на сроки разработки, технологию проектирования корректировок комплекса. К факторам, влияющим на ***сложность обнаруживаемых ошибок*** комплексов программ, относятся:

- величина – размер создаваемой или модифицируемой программы, выраженная числом строк текста, функциональных точек или количеством программных компонентов в комплексе;
- количество обрабатываемых переменных или размер и структура памяти, используемой для размещения базы данных корректировок;
- трудоемкость разработки изменений компонентов и комплекса программ;
- длительность разработки и реализации корректировок;
- число специалистов, участвующих в производстве компонентов и комплекса программ.

Убывание ошибок в компонентах и комплексе программ и интенсивности их обнаружения ***в процессе производства не беспредельно***. После тестирования в течение некоторого времени интенсивность обнаружения дефектов при самых жестких внешних условиях испытаний снижается настолько, что коллектив, ведущий разработку и тестирование, попадает в ***зону нечувствительности к ошибкам и возможным отказам функционирования***. При такой интенсивности отказов вследствие их редкого проявления трудно прогнозировать затраты времени, необходимые для обнаружения очередной ошибки или дефекта. Создается представление о полном отсутствии случайных проявлений дефектов, о невозможности и бесцельности их поиска, поэтому усилия на тестирование сокращаются, и интенсивность обнаружения ошибок еще больше снижается. Этой ***предельной интенсивности обнаружения отказов*** соответствует наработка на обнаруженную ошибку, при которой ***прекращается улучшение характеристик программного комплекса*** на этапах тестирования или испытаний.

При серийном выпуске программного продукта, благодаря значительному расширению вариантов исходных данных и условий эксплуатации, возможно в течение некоторого времени возрастание суммарной (по всем экземплярам системы) интенсивности обнаружения дефектов и ошибок. Это позволяет дополнительно устранять ряд дефектов и тем самым увеличивать длительность между проявлениями ошибок в процессе эксплуатации.

Глава 2.2

ОРГАНИЗАЦИЯ ВЕРИФИКАЦИИ И ТЕСТИРОВАНИЯ КОМПОНЕНТОВ И КОМПЛЕКСОВ ПРОГРАММ

Процессы верификации компонентов и комплексов программ

Широко распространенная ***неупорядоченная разработка требований*** к функциям, процедурам и взаимодействию программных компонентов не может гарантировать высокое качество сложных программных продуктов. При этом пропускаются и не проверяются некоторые сценарии и маршруты исполнения программ, которые реализуют важнейшие требования контракта и исходных спецификаций. Удовлетворение потребностей заказчика является целью любого программного проекта. Соответственно, ***обеспечение качества, полноты и корректности требований*** в проекте невозможно представить без адекватных процессов работы с ними – начиная с ***верификации требований***, заканчивая проверкой соответствия получаемого программного комплекса этим требованиям на всех этапах его создания.

Верификация – это процесс для определения, выполняет ли программный комплекс и его компоненты требования, наложенные на них в последовательных этапах ЖЦ КП. Анализ, просмотры (обзоры) и тестирование взаимодействия требований являются важнейшей частью верификации и установления корректности программ. Основная цель верификации КП состоит в том, чтобы обнаружить, зарегистрировать и устранить дефекты и ошибки, которые внесены во время последовательной разработки требований или модификации программных комплексов и компонентов. Для эффективности затрат ресурсов при ее реализации, верификация должна быть интегрирована как можно раньше с процессами проектирования, производства и сопровождения. Обычно она проводится сверху вниз, начиная от общих требований, заданных в техническом задании и/или спецификации на всю систему до детальных требований на программные модули и их взаимодействие – рис.2.2.

Основные задачи верификации систем и программных комплексов включают:

- верификацию договора с заказчиком на цели и функции проекта по критериям:
  - требования договора непротиворечивы и покрывают все потребности заказчика и пользователей;
  - предусмотрены критерии и процедуры приемки результатов разработки на соответствие требованиям договора;
- верификацию требований к функциям системы по критериям:
  - требования к системе непротиворечивы, выполнимы, тестируемы и проверяемы;
  - требования к программному комплексу последовательны, тестируемы и точно отражают требования системы;
- верификацию требований к модулям, компонентам и комплексу программ по критериям:
  - тексты программных модулей и компонентов удовлетворяют требованиям к комплексу программ, тестируемы, корректны и соответствуют стандартам программирования;
  - компоненты и комплекс программ корректно обеспечивают надежность, безопасность, производительность и другие критические требования;
- верификацию интеграции модулей, компонентов и комплекса программ по критериям:
  - модули и компоненты полностью и правильно могут быть интегрированы в комплекс программ;
  - компоненты аппаратных средств, программного продукта и ручные операции полностью и правильно интегрированы в систему;
- верификацию документации комплекса программ по критериям:
  - эксплуатационная и технологическая документация компонентов и комплекса программ адекватна, корректна, полна и непротиворечива;
  - управление конфигурацией документов соответствует плановым процедурам производства.

Рис. 2.2.

Назначение верификации – последовательно проверить, что в реализуемом комплексе программ[20, 33]:

- общие требования к информационной системе, предназначенные для программной реализации, корректно переработаны в спецификацию требований высокого уровня к комплексу программ, удовлетворяющие исходным системным требованиям;

- требования высокого уровня правильно переработаны в архитектуру КП и в спецификации требований к функциональным компонентам низкого уровня, которые удовлетворяют требованиям высокого уровня;
- спецификации требований к функциональным компонентам КП, расположенным между компонентами высокого и низкого уровня, каждый раз удовлетворяют требованиям более высокого уровня;
- архитектура КП и спецификации требований к компонентам низкого уровня корректно переработаны в, удовлетворяющие им, исходные тексты программных и информационных модулей;
- исполняемый объектный код удовлетворяет требованиям к исходному тексту программных компонентов.

Кроме того, верификации на соответствие спецификации требований на конкретный проект программного продукта подлежат требования к эксплуатационной и технологической документации.

Цели верификации комплекса программ достигаются посредством последовательного выполнения комбинации из просмотров, анализов, разработки тестовых сценариев и процедур и последующего выполнения этих процедур. Выполнение тестовых процедур должно обеспечивать демонстрацию соответствия испытываемых программ исходным требованиям. Результаты верификации должны быть включены в документы: выполненные процедуры верификации, описание и отчет о квалификационном тестировании комплекса и его компонентов.

Верификацию следует проводить постоянно, чтобы убедиться, что каждый компонент и модуль корректен, удовлетворяет потребности следующего компонента и не содержит лишних или неопределенных процедур. Важным показателем качества реализации компонента и комплекса программ является возможность трассировки требований на последовательных этапах спецификации, архитектуры, проектирования, реализации и тестирования. Способность отслеживать ***отношения между требованиями*** и учитывать их связи при возникновении изменений является основой многих современных высоконадежных программных процессов, особенно в критических областях создания и применения заказных программных продуктов [2, 3, 44].

Верификация комплекса и компонентов программ – это процесс, предназначенный для определения, выполняют ли программный комплекс, его компоненты, модули, требования и условия, наложенные на них в предыдущих этапах жизненного цикла комплекса про-

грамм. Для эффективности затрат при ее реализации, верификация должна быть интегрирована как можно раньше с процессами проектирования и производства программ. Эти процессы включают анализ, трассирование и тестирование выполнения требований, которые являются важнейшими компонентами верификации. Они проводятся ***сверху вниз***, начиная от общих требований в техническом задании и/или спецификации на всю систему до детальных требований на отдельные компоненты, модули и их взаимодействие.

Просмотры и анализы требований высокого уровня предназначены для того, чтобы обнаруживать, регистрировать и устранять дефекты и ошибки, которые внесены в процессе последовательной разработки и декомпозиции требований к системе. Эти просмотры и анализы должны подтвердить корректность и согласованность системных требований высокого уровня, а также гарантировать, что:

- полностью определены функции системы, которые должен выполнять программный продукт;
- требования по функциональности, эффективности и к качеству системы детализированы в исходных требованиях высокого уровня к комплексу программ и что правильно определены производные требования и обоснована их необходимость;
- каждое требование высокого уровня к программному продукту является точным, однозначным и достаточно детализированным, и что требования не конфликтуют друг с другом;
- не существует никаких конфликтов между требованиями высокого уровня и возможностями аппаратных и программных средств объектного компьютера, особенно такими, как время реакции системы и характеристики аппаратуры ввода/вывода;
- процесс разработки требований к программному продукту полностью соответствует стандартам на создание спецификаций требований и любые отклонения от стандартов обоснованы;
- функциональные и конструктивные характеристики качества, предназначенные для программной реализации, полностью включены в требования высокого уровня к КП.

Просмотры и анализы исходных текстов программ предназначены для выявления и регистрации дефектов и ошибок, которые внесены в процессе программирования компонентов. Эти процедуры должны подтверждать, что выходные результаты кодирования алгоритмов – тексты программ, являются точными, полными и могут быть проверены. Прежде всего, должна определяться корректность

текста программ по отношению к исходным требованиям и к архитектуре комплекса, и соответствие стандартам на программирование. Эти просмотры и анализы обычно ограничиваются исходным текстом программ, которые подтверждают, что он является корректным, полным и соответствует требованиям низкого уровня к компонентам, а также то, что исходный текст не содержит излишних и неописанных функций. Должно быть проверено, что процесс разработки программ осуществлялся в соответствии со стандартами на программирование и отклонения от этих стандартов обоснованы, особенно в случаях ограничения сложности и использования конструкций программ, которые должны удовлетворять целям корректности системы. Сложность в данном контексте понимается как ***степень связности*** программных компонентов, уровень вложенности управляющих структур и сложность логических и числовых выражений.

Для обеспечения высокого качества программного продукта параллельно с ***верификацией требований*** и программированием корректировок, целесообразно разрабатывать и верифицировать ***спецификации и сценарии тестов***, отражающие методы и конкретные процедуры проверки реализации изменений этих требований (рис. 2.3). Тестовые спецификации могут использоваться для проверки согласованности, внутренней непротиворечивости и полноты реализации требований без исполнения программ. Для каждого требования к комплексу программ, его архитектуре, функциональным компонентам и модулям должна быть разработана спецификация тестов, обеспечивающая проверку корректности, адекватности и возможности в последующем реализовать тестирование компонента на соответствие этому требованию. Такая взаимная проверка функций компонентов, отраженных в требованиях и в спецификациях тестов, обеспечивает повышение их качества, сокращение дефектов, ошибок, неоднозначностей и противоречий [35, 44].

При тестировании программ зачастую оказывается, что не каждое требование в спецификации описано достаточно полно и корректно, чтобы оно могло быть проверено тестами. При разработке тестов на основе таких спецификаций вследствие их возможной неопределенности, может обнаруживаться, что ***не для каждого требования*** к программе или данным может быть подготовлен тест.

С другой стороны ***не для каждого теста*** может оказаться в спецификации адекватное требование на функции компонентов и КП. Спецификации тестов должны обеспечивать дополнительный кон-

троль корректности требований и верификацию взаимодействия компонентов на соответствующем уровне описания КП. Независимая разработка тестов на основе спецификаций требований, создает базу для обнаружения требований, которые не тестировались или принципиально не могут быть проверены тестированием. Таким образом, ***верификация спецификаций требований на*** программные компоненты и КП могут использоваться с ***двумя целями*** (см. рис.2.3):

- для разработки, программирования и проверки текстов программ и интерфейсов взаимодействия программных компонентов разных уровней в комплексе программ;
- для создания ***скоординированного комплекса тестов*** для совокупности компонентов, обеспечивающих взаимную проверку реализации спецификаций требований на комплекс программ и компоненты.

В результате совокупности требований к тестам могут применяться при разработке и сопровождении ***как эталоны и вторая адекватная форма описания содержания программ*** для сквозной верификации спецификаций требований к тестам сверху вниз, а также сами подвергаться верификации на корректность соответствия исходным требованиям к компонентам текстов программ и данных разных уровней. Такие ***параллельные взаимные проверки*** спецификаций требований и текстов программ, и спецификаций тестов способствуют выявлению и исключению множества вторичных дефектов и ошибок. Впоследствии эти спецификации тестов должны использоваться для непосредственного тестирования исполнения требований к программным компонентам соответствующего уровня. Кроме того, параллельная и независимая разработка, с одной стороны, спецификаций программ и спецификаций тестов, а также их реализации, с другой стороны, позволяет распараллеливать работы в ЖЦ, что ведет к сокращению сроков создания компонентов и комплексов программ.

Реализация ***взаимодействия верификации и тестирования*** может производиться разными методами и независимыми специалистами – программистами и тестировщиками, что позволяет использовать результаты их деятельности для сравнения содержания одних и тех же программ, представленных на языках программирования и описанных на языках тестов.

Рис. 2.3.

Особенности описаний и реализации программ, а также ***мышления программистов*** – на основе функций и процедур исполнения программ, существенно ***отличаются от представлений и методов описаний тех же функций программ тестировщиками*** – создателями сценариев тестирования. Они акцентируют деятельность на конкретных процедурах проверки функционирования, возможных результатах и взаимодействии компонентов КП. Это позволяет выявлять вторичные дефекты, появляющиеся при корректировках, и повышать качество производства и сопровождения путем сопоставления двух методов и результатов описания одних и тех же программ, за счет того, что мала вероятность одинаковых ошибок в сценариях тестов и в реализации текстов программ [35, 44].

Чтобы принять решение о том, какая часть системы нуждается в верификации и проверке корректности и в каком объеме, целесообразно применять ***анализ и оценку рисков***. Это позволяет определить,

для каких компонентов требований их неправильная реализация недопустима, а также разработать план действий по верификации и проверке правильности, основываясь на результатах этих оценок. На этом этапе следует привлекать к управлению требованиями и анализу их корректности, специалистов по тестированию, подключая их к планированию тестов с самого начала проекта. Группа тестирования должна разрабатывать тестовые процедуры и сценарии, которые трассируются к прецедентам, а также функциональным и конструктивным требованиям.

Если заказчик проекта требует ***гарантировать независимую верификацию*** всего комплекса требований, то должна быть выбрана квалифицированная организация, ответственная за выполнение полной верификации. Этой организации должны гарантироваться независимость и полномочия верификационной деятельности, по обеспечению корректности совокупности требований. Должны быть определены плановые действия жизненного цикла системы, компоненты и программные продукты, требующие верификации. Дефекты и несоответствия, обнаруженные при верификации, должны быть введены в ***процесс корректировки требований к комплексу программ***. Результаты верификационной деятельности должны быть доступны заказчику и потребителю системы и программного продукта (см. рис. 2.2).

В стандартах на жизненный цикл при выполнении работ по верификации комплекса программ рекомендуется следующая ***последовательность основных процедур***:

- должны быть проверены корректность и непротиворечивость исходных требований высокого уровня (контракта) и продемонстрирована трассируемость и прослеживаемость к ним последующих требований;
- результаты анализа прослеживаемости и анализа покрытия исходных требований и структуры должны показывать, что каждое требование к комплексу программ ***является трассируемым*** к компонентам и объектному коду модулей, которые его реализуют, а выполненные просмотры, анализы и сгенерированные тестовые наборы способны проверить эти требования;
- когда невозможно проверить некоторые требования посредством исполнения программ в тестовой среде адекватной реальной, должны быть обеспечены другие средства, и в документах должна быть обоснована возможность их использования для удовлетворения целей верификации комплекса программ;

- несоответствия и ошибки, обнаруженные в процессе верификации, должны быть зарегистрированы для последующего исследования и корректировки компонентов, комплекса программ и/или требований.

Верификация, просмотры и анализы требований высокого уровня предназначены обнаруживать, регистрировать и устранять ошибки, которые внесены в процессе разработки и преобразования требований к комплексу программ. Эти просмотры и анализы должны подтвердить корректность и согласованность требований высокого уровня, а также ***гарантировать***, что:

- полностью определены функции системы, которые должен выполнять программный продукт, а также требования к характеристикам по функциональности, эффективности, надежности и безопасности системы, правильно определены производные требования и обоснована их необходимость;
- каждое требование высокого уровня к программному комплексу является точным, однозначным и достаточно детализированным, требования не конфликтуют друг с другом;
- не существует конфликтов между требованиями высокого уровня и возможностями аппаратных и программных средств реализующего компьютера, особенно такими, как время реакции системы и аппаратуры ввода/вывода;
- каждое требование высокого уровня к программному комплексу может быть проверено и верифицировано;
- процесс разработки требований к комплексу программ полностью соответствует стандартам на создание требований и обоснованы любые отклонения от стандартов;
- системные, функциональные требования, требования по эффективности, надежности и безопасности системы, предназначенные для программной реализации, отражены в требованиях высокого уровня.

Верификация, просмотры и анализы требований архитектуры программного комплекса предназначены обнаруживать и регистрировать дефекты и ошибки, которые могли быть внесены во время разработки архитектуры комплекса программ. Эти просмотры и анализы должны подтвердить, что архитектура комплекса программ не находится в противоречии с требованиями высокого уровня, выполняются все функции, которые гарантируют целостность системы, а

также существуют корректные связь между функциональными компонентами архитектуры комплекса.

Верификация, просмотры и анализы требований к программным компонентам и модулям предназначены выявлять дефекты и ошибки детализированных требований, которые внесены в процессе проектирования функциональных компонентов. Эти анализы должны гарантировать, что требования низкого уровня к компонентам и модулям удовлетворяют требованиям высокого уровня, и что правильно определены формируемые из них требования и обоснована их необходимость. Следует гарантировать, что каждое требование низкого уровня является точным, однозначным и достаточно детализированным ***для программирования модуля*** и что требования низкого уровня не конфликтуют друг с другом.

Верификация, просмотры и анализы исходного текста модулей программ должны выявлять и регистрировать ошибки, которые внесены в процессе программирования модулей и компонентов комплекса. Они должны подтверждать, что выходные результаты программирования модулей являются точными, полными и могут быть верифицированы. Прежде всего, должна проверяться корректность текста программ по отношению к исходным требованиям к архитектуре комплекса программ. Анализы обычно ограничиваются исходным текстом модулей программ, проверяется, что он является корректным, полным и соответствует требованиям к соответствующим компонентам низкого уровня, а также то, что исходный текст не содержит не описанных функций. Должно быть проверено, что процесс разработки программ осуществлялся полностью в соответствии с ***верифицированными требованиями к компонентам и модулям***, а также ***по стандартам на программирование***, и отклонения от этих стандартов обоснованы, особенно, в случаях ограничения сложности и использования конструкций программ, которые должны удовлетворять целям безопасности системы.

Верификация, просмотры и анализы тестовых вариантов, процедур и результатов должны показать, что требования к тестированию комплекса, компонентов и модулей разработаны и могут выполняться точно и полностью (см. рис. 2.3). Должно быть рассмотрено и подтверждено верификацией, что тестовые сценарии и варианты исходных данных правильно представлены в процедурах тестирования и ожидаемых результатах; гарантируют, что составы результатов тестирования требований являются корректными и что

объяснимы расхождения между фактическими и ожидаемыми результатами.

В той или иной степени оценкой затрат на верификацию приходится заниматься на ***каждой*** стадии производства. Она необходима, чтобы удостовериться в правильности и затратах на переходы:

- от потребностей заказчика и пользователя к функциям программного комплекса;
- от требований к комплексу программ к его архитектуре;
- от архитектуры к модели проектирования и производства модулей, компонентов и комплекса программ;
- от реализации модулей, компонентов и комплекса программ к планированию и реализации совокупности тестов (см. рис. 2.2).

Особое внимание следует уделять этапу ***завершения проекта***. В успешном процессе это еще один шаг – испытания, подтверждающий, что система сконструирована и реализована в соответствии ***с потребностями заказчика***. Кроме того, на этом этапе необходимо показать, что система действительно работает так, как требовалось.

Трассирование взаимодействия требований к компонентам в комплексах программ

Поддержка верификации осуществляется путем использования методов ***трассировки требований***, что позволяет связывать друг с другом части проекта, проводить проверку адекватности требований к прецедентам их реализации и функциям, и обратно. С помощью трассировки можно удостоверяться в том, что[34, 44]:

- все компоненты исходных требований проекта учтены;
- все реализуемые компоненты служат заданной цели и требованиям к комплексу программ.

Посредством ***трассировки*** следует устанавливать корректность связей между двумя или большим числом ***компонентов и/или процессов разработки требований***, которые являются: предшествующими – последующими, или главными – подчиненными, а также соответствие между требованиями и их реализацией конкретными программными компонентами. Каждый компонент и модуль программного комплекса должен ***оправдывать свое существование и соответствовать каким-то заданным требованиям***. Ключевыми элементами верификации и тестирования являются отношения трассировки. Эти отношения можно определять с помощью модели, использующей понятия ***«трассируется к»*** и/или ***«трассируется от»***. Если одно или несколько требований к программному компоненту

создаются с целью поддержки некоторой функции, заданной в исходном документе, то требование трассируется ***от*** некоторой функции. Если некоторое требование к программному компоненту «трассируется ***к***» определенному тестовому сценарию, то данное требование тестируется этим скриптом. То, что описание компонента «трассируется ***от***» конкретного программного требования, подразумевает, что это требование реализуется указанным компонентом.

Потребности заказчика должны отслеживаться путем ***анализа содержания требований***, чтобы можно было определить, какие требования будут затронуты, если в течение или после разработки, потребности изменятся. Это также дает уверенность, что в спецификации требований указаны все потребности заказчика. Кроме того, можно проследить ***в направлении от требований*** к потребностям заказчика, чтобы определить происхождение каждого требования к комплексу программ. Если необходимо представить потребности заказчика в форме сценариев использования функций, то анализ должен отражать ***трассирование между вариантами использования*** и функциональными требованиями.

По мере производства комплекса программ, процессы можно отслеживать ***в направлении от требований***, и определять связи между отдельными требованиями и компонентами комплекса. Этот тип связей гарантирует, что каждое требование удовлетворено, поскольку установлено, какой компонент соответствует этому требованию. Еще один тип связей может контролировать отдельные элементы продукта в ***направлении к требованиям*** для того, чтобы знать ***причину и цель*** создания каждого компонента. В большинстве комплексов программ могут быть компоненты, не относящиеся, напрямую, к требованиям заказчика, но необходимо устанавливать, ***для чего нужен каждый компонент***.

Если трассировщик обнаружит незапланированную функциональность при отсутствии соответствующего требования, то фрагмент программы может свидетельствовать, что разработчик реализовал требование, которое аналитик или заказчик может добавить к спецификации. Однако это может быть элемент программы, ***«украшающий»*** фрагмент, который не относится к комплексу. Связи трассируемости помогут сортировать подобные ситуации и получать более полное представление о том, как именно компоненты системы ***составляют целое, соответствующее требованиям***. Сценарии верификации или тестирования, которые созданы на основе отдельных

требований, которые можно проследить до этих требований, представляют собой механизм ***выявления нереализованных требований***, поскольку нет ожидаемой функции или компонента. Пропуск реализации, верификации и тестирования требования – может быть существенным дефектом, если заказчик не удовлетворен или в готовом продукте отсутствует функция, особо важная для обеспечения надежности или безопасности.

Трассирование требований сложного комплекса программ – трудоемкая задача, обычно выполняемая вручную, для которой необходима соответствующая организация и квалификация специалистов. Если в ходе разработки тщательно фиксируются данные трассируемости требований, у руководителей будет точное представление о состоянии реализации запланированной функциональности и характеристик программного комплекса. Отсутствующие связи от требований указывают на компоненты, которые еще не созданы. Если тестирование дает неожиданный результат, то трассирование связей ***между тестами, требованиями и текстом модулей и компонентов*** могут указать на наиболее вероятные части программного кода, которые необходимо проверить на наличие дефектов. Информация о том, какие тесты проверяют какие требования, экономит время, позволяя удалять лишние, выявлять и создавать необходимые тесты (см. рис. 2.4).

Информация трассируемости облегчает внесение изменений в ходе сопровождения, что повышает ***производительность разработчиков при модификации комплекса программ***. Информацией трассируемости целесообразно пользоваться при сертификации продукта с особыми требованиями к надежности и безопасности, чтобы продемонстрировать заказчику, что все требования были реализованы, хотя это не доказывает, что они реализованы корректно и полностью. Естественно, если требования некорректны или отсутствуют ключевые требования, то результаты трассируемости не помогут.

Документирование взаимосвязей компонентов ***уменьшает риск возникновения проблем***, если вдруг ключевой член команды, обладающей важной информацией о системе, покидает проект. Отношения трассировки между компонентами проекта могут быть явными или неявными. ***Явная трассировка*** – связь или отношение, между функцией комплекса и компонентом, осуществляющим поддержку этой функции, которая определяется исключительно решением специалиста о том, что такое отношение имеет смысл.

Рис. 2.4.

Методология разработки и структура системы могут определять ***неявные отношения трассировки – «дочерних»*** требований между компонентами и ***«родительскими»*** требованиями, когда существуют формальные, иерархические отношения. Связи трассируемости помогают отслеживать «родительские» требования, взаимосвязи и зависимости между отдельными требованиями. Эта информация отражает влияние изменения, если отдельное требование удаляется или модифицируется.

Удобный способ представления связей между требованиями и другими компонентами системы – ***матрица трассируемости требований***. Каждое функциональное требование в такой матрице, связано с определенном вариантом использования (в направлении «назад»), и с одним или более, элементами верификации и тестирования (в направлении «вперед»). Можно добавить дополнительные столбцы для расширения ссылок на другие рабочие продукты, например, на документацию системы. После того как с помощью инструментального средства заданы все известные отношения между компонентами, обязательным действием является проверка матрицы трассиров-

ки взаимосвязей компонентов на наличие следующих ***двух возможных индикаторов дефектов или ошибок***.

Если при просмотре некой ***строки*** матрицы связей не удается обнаружить никаких отношений трассировки, вероятно, что еще не определено требование к программному компоненту, отвечающее функции исходного документа требований. Тем не менее, пустые строки являются индикаторами возможных ошибок и нуждаются в тщательной проверке. Современные средства управления требованиями должны предоставлять возможность автоматизированного проведения такой проверки.

Если в некотором ***столбце не оказывается*** отмеченных отношений трассировки, вероятно, было создано требование к программному компоненту, для которого нет требующей его, функции продукта. Это может указывать на неправильное понимание роли программного требования, недостаток исходного документа проекта, а также на то, что компонент программы неправильный, не соответствует системному требованию или является дефектом разработчика, и в таком случае его следует удалить.

Чтобы обнаружить ***пропущенные отношения***, надо искать ***строки матрицы трассировки***, которые показывают, что некая функция ***не*** связана ни с одним программным требованием (прецедентом). При обнаружении пропуска в отношениях нужно вернуться к исходному набору требований к комплексу программ и связанным с ними программным требованиям.

Организация процессов тестирования компонентов и комплексов программ

Многолетний опыт показал, что единственный универсальный метод тестирования сложных комплексов программ создать невозможно и следует применять упорядоченный ряд значительно различающихся методов. Каждый метод отличается целевыми задачами тестирования, проверяемыми компонентами программы и методами оценки результатов [9, 12, 23]. ***Только совместное и систематическое применение различных методов тестирования позволяет достигать высокое качество функционирования сложных комплексов программ***. Традиционная **V – модель,** отражает организацию, взаимодействия процессов формирования и декомпозиции требований к функциям, характеристикам и структуре комплексов программ, и последовательного комплексирования и тестирования для удовлетворения требований – рис. 2.4. Анализ, проектирование и

производство представляются ***сверху вниз*** левой стороной V – модели, а сборка, тестирование и испытания компонентов ***снизу вверх*** – правой стороной схемы. Эта модель иллюстрирует естественные потоки и взаимодействие процессов разработки требований и тестирования при последовательной реализации основных базовых элементов ЖЦ сложных программных комплексов, компонентов и модулей. При этом важную роль играет адекватная организация коллектива специалистов реализующих процессы проектирования, производства и тестирования [35, 44].

Основой процессов организации тестирования являются ***требования к функциям и характеристикам заказной системы*** управления и обработки информации, использующей программный продукт. ***Системные, функциональные требования*** – должны отражать потребности к задачам и функциям программного комплекса, содержащего взаимосвязанные компоненты и модули. При этом, система может быть как целиком программной, так и состоять из программной и аппаратной частей. В общем случае, частью системы может быть оперативный персонал, выполняющий определенные функции системы. Программные комплексы все чаще встраиваются в различные заказные системы. Работа над такими проектами требует коллектива квалифицированных программных специалистов имеющих широкий взгляд на общие ***задачи проектирования систем***. Программным менеджерам и аналитикам необходимо участвовать в выработке требований для всей системы, а также ***освоить прикладную область***, требованиям которой должен отвечать программный продукт.

Спецификации требований к системе являются результатом системных решений и источником для соглашений о производстве или приобретении функциональных программных компонентов и критериев их приемки. Они также являются основой для принятия решений по производству или приобретению и повторному использованию компонентов, для их верификации и для установления стратегии комплексирования этих компонентов в сложный комплекс программ. Порядок проектирования архитектуры должен позволять проводить анализ изменений требований в течение ЖЦ системы, а также является источником информации для последующего повторного использования архитектуры и компонентов. Также это источник информации, при помощи которой ***определяются необходимые тесты*** в ходе комплексирования компонентов системы, которые ниже пред-

ставлены процессами V – модели с краткими комментариями [9, 12, 23].

Требования к функциям, характеристикам и структуре программного комплекса формируются в модели на основе заданных функций, свойств и характеристик системы, и должны удовлетворить возможность его применения по основному назначению с требуемым качеством. Они должны решать весь комплекс задач системы с учетом ограниченных ресурсов (см. рис. 2.4).

Декомпозиция требований и структуры программного комплекса на программные компоненты необходима для снижения сложности и реструктуризации компонентов комплекса, позволяющей вести распределенную разработку и тестирование программ относительно небольшими группами специалистов, которым представляются корректные требования к функциям, характеристикам и качеству определенных компонентов.

Декомпозиция требований и структуры программных компонентов на программные модули должна обеспечивать возможность корректной формализации требований к функциям и характеристикам каждого модуля, тестируемого определенным тестировщиком, гарантирующим его качество и возможность применения в определенных программных компонентах.

Производство (и/или выбор готовых) и тестирование программных модулей на соответствие с требованиям, должны гарантировать их качество и обеспечивать возможности включения в программные компоненты. Модули являются наиболее массовыми элементами комплексов программ, в их создании участвуют наименее квалифицированные специалисты и при тестировании выявляется наибольшее количество дефектов и ошибок.

Сборка, комплексирование модулей и тестирование компонентов на соответствие требованиям должны сформировать и подтвердить реализацию функций достаточно крупными функциональными компонентами высокого качества, готовыми к встраиванию в комплекс программ.

Сборка, комплексирование компонентов и тестирование комплекса программ на соответствие требованиям должны завершить тестирование комплекса с полным набором компонентов, соответствующих требованиям, и реализацию всех функций программного комплекса.

Испытания программного комплекса на соответствие требованиям к функциям, характеристикам и структуре должны обеспечить корректное решение и контроль всех функциональных задач в соответствие с требованиями к системе, пригодность для поставки и внедрению в систему заказчика и пользователям.

При выборе и применении методов ***тестирования программных компонентов*** необходимо учитывать общие требования к ним и следующие их ***особенности***:

- относительно высокая доля творческого труда специалистов, осуществляющих тестирование, приводит к необходимости обеспечения высокоэффективного интерактивного их взаимодействия со средствами автоматизации проверок;
- непредсказуемость видов и мест выявляемых ошибок в программах ограничивает возможность автоматического их обнаружения и определяет необходимость ориентироваться на автоматизированные методы и средства при творческой, высокой роли специалистов при тестировании;
- разнообразие возможных мест расположения и видов ошибок, при относительно редком их обнаружении, приводит к регистрации и анализу большого объема избыточной информации о процессе исполнения программ при тестировании;
- высокая сложность отлаживаемых программных компонентов, творческий и итерационный характер процесса тестирования затрудняют и ограничивают возможную точность оценки полноты проведенного тестирования и достигнутого качества компонентов;
- активное участие в тестировании специалистов, различающихся по квалификации, опыту, темпераменту и творческим возможностям, а также различия архитектуры, сложности и функций отлаживаемых программных компонентов не позволяют жестко регламентировать трудоемкость, методики и технологии применения видов и средств автоматизации тестирования.

В процессе производства программ специалисты должны стремиться охватить тестированием функционирование комплекса программ во всей доступной области изменения исходных данных и режимов применения. При этом номенклатура и диапазоны варьирования тестов ограничены либо допустимой длительностью подготовки и исполнения тестов, либо вычислительными ресурсами, доступными для использования с целью контроля и тестирования в режиме нормальной эксплуатации. На выбор эффективных методов тестирования

и последовательность их применения в наибольшей степени влияют основные ***характеристики заказных тестируемых объектов***:

- класс заказных комплексов программ, определяется глубиной связи его функционирования с реальным временем и случайными воздействиями из внешней среды, а также жесткими требованиями заказчика к качеству обработки информации и надежности функционирования программ;
- сложность или масштаб (размеры) комплекса программ и его функциональных компонентов, являются конечными результатами производства;
- преобладают следующие элементы в программах: осуществляющие вычисления сложных выражений и преобразования измеряемых величин или обрабатывающие логические и символьные данные для подготовки и отображения функциональных решений [20, 33].

Анализ покрытия тестами требований к комплексу программ должен определять, какие требования не были протестированы и какие структуры программного комплекса не были исполнены при тестировании. Оценка тестового покрытия состоит из ***двух шагов***, включающих анализ покрытия, основанного на требованиях, и анализ структурного покрытия. ***Первый шаг*** анализирует тестовые наборы относительно требований к комплексу программ, чтобы подтвердить, что выбранные наборы тестов удовлетворяют установленным критериям. Этот анализ покрытия, основанного на требованиях, должен определить, насколько полно при тестировании проверена реализация всех требований в спецификации, и показать потребность в дополнительных тестовых наборах. Тестовые варианты, основанные на требованиях, могут не полностью покрыть структуру программы. Поэтому дополнительно целесообразно выполнять анализ структурного покрытия и проводить его верификацию. ***Второй шаг*** должен подтвердить, что процедуры тестирования, основанные на требованиях, покрыли всю структуру программы. Анализ структурного покрытия должен определять, не пропущены ли элементы структуры программы, которые не проверены тестовыми процедурами, основанными на требованиях.

Анализ начинается с основных методов тестирования и обеспечения ***качества относительно простых программ***. К ним относятся отдельные программные модули (ПМ) или их небольшие группы, решающие достаточно простые функциональные задачи. Однако ниже эти компоненты не различаются, и преимущественно ***использует-***

ся термин – программный модуль – ПМ. Эти объекты тестирования имеют ряд принципиальных особенностей, которые отличают их от сложных программных комплексов и непосредственно отражаются на применяемых методах и средствах автоматизации тестирования. Невысокая размерность ПМ обеспечивает их обозримость и возможность детального анализа функций, структуры программы и процесса решения задач. Детализированный контроль ПМ должен проводиться планомерно и детерминировано, с точностью до любого оператора или операнда в программе. Стремление к снижению сложности тестирования реализуется частично путем укрупнения наблюдаемых и анализируемых компонентов до уровня небольших групп операторов на языке программирования в пределах одного структурного элемента программы. В результате при тестировании имеется возможность оценивать и контролировать степень проведенной проверки и достигнутое качество компонентов, а также выделять следующие виды тестирования [2, 3, 44].

Тестирование для обнаружения ошибок имеет целью выявление отклонений результатов функционирования реальной программы от заданных требований и эталонных значений. Задача состоит в обнаружении максимального числа дефектов программы, в качестве которых принимается любое отклонение результатов от эталонов. Успешным является тестирование, которое приводит к обнаружению существования ошибок при минимальных затратах.

Тестирования для диагностики и локализации ошибок предназначено для того, чтобы точно установить исходное место искажения программ или данных, являющегося причиной и источником отклонения результатов от эталонов при тестировании для обнаружения ошибок. Эффективными являются тесты, способствующие быстрой и точной локализации первичных дефектов программ и данных. На этой стадии затраты оправданы и тестирование можно считать успешным, если оно привело к определению элементов программы или данных, подлежащих непосредственной корректировке.

Методы и стратегии тестирования программных компонентов используются для обработки текстов программ в различной форме и для представления информации о результатах тестирования в виде удобном для анализа специалистами. Форма представления и исполнения программы для тестирования зависит от этапов разработки, уровня языков программирования, наличия соответствующих средств автоматизации и других факторов. Программы в процессе

тестирования используются в ***двух принципиально различных формах представления:***

- в виде текста ***на языке программирования*** или формализованного описания спецификаций требований (символьное представление), удобного для анализа человеком и обычно недоступного для непосредственного получения результатов функционирования программы в компьютере;
- в ***машинном коде*** конкретного компьютера (объектное представление), пригодном для автоматической обработки определенных кодовых исходных данных и не всегда удобном для их анализа человеком.

Первичные и вторичные дефекты и ошибки в программах выявляются на последовательных этапах тестирования. Вторичные ошибки являются определяющими для качества функционирования программ, так как не каждая первичная ошибка вносит заметный вклад в искажение выходных результатов. Существенной особенностью тестирования ПМ является относительная ***близость проявления вторичных ошибок к их причинам – первичным ошибкам***. Это означает, что последствия ошибок обнаруживаются в искажениях результатов непосредственно в месте локализации первичной ошибки или после небольшого числа шагов исполнения программы.

Анализ программы на уровне символов и операторов языка программирования обеспечивает все виды тестирования: для обнаружения ошибок, для их локализации и для проверки правильности выполненных корректировок. Эта особенность используется не только при автономной отладке модулей, но и при комплексном тестировании и обнаружении дефектов функционирования КП. В последнем случае приходится итерационно углубляться в проверяемые компоненты, вплоть до модуля и его операторов. При этом для локализации и исправления первичной ошибки в ряде случаев, приходится возвращаться к тестированию модулей на уровне операторов программы.

Еще одной особенностью тестирования ПМ является необходимость и возможность обеспечения гарантий их высокого качества и пригодности для использования в разных проектах. ***Перспективы многократного использования апробированных программных компонентов*** в различных версиях КП могут быть обеспечены только при четкой формализации их функций и условий применения, а также при доверии к корректной реализации функций и интерфейсов в оп-

ределенной среде. Завершать отладку и испытания ПМ и компонентов следует их аттестацией, с приложением перечня реализуемых требований, интерфейсов, характеристик полноты тестирования и диапазонов варьирования тестовых данных. Такая аттестация может служить гарантией качества для многократного использования компонентов в различной аппаратной и операционной среде.

Тестирование потоков управления (структуры программы) должно быть начальным этапом, так как при некорректной структуре возможны наиболее грубые искажения выходных результатов и даже отсутствие некоторых из них. Оно состоит в проверке корректности последовательностей передач управления и формирования маршрутов исполнения компонентов программы при обработке тестов. Для тестирования структуры программ в большинстве случаев требуются относительно меньшие затраты по сравнению с тестированием на потоках данных.

Тестирование потоков данных можно разделить на два этапа. Первый этап тестирования состоит в анализе обработки данных, определяющих значения предикатов в операторах выработки логических решений. Эти решения влияют на маршруты обработки информации, что сближает в этой части метод тестирования потоков данных с тестированием структуры программы. Второй этап тестирования обработки данных состоит в проверке вычислений по аналитическим формулам или численных значений результатов в зависимости от числовых или логических значений исходных данных. В качестве эталонов используются результаты ручных или автоматизированных расчетов по тем же или близким по содержанию формулам.

Любые методы тестирования ПМ и компонентов, в большей или меньшей степени, ориентированы на обнаружение ошибок определенных типов. Методы тестирования потоков управления предназначены преимущественно для обнаружения ошибок в структуре компонентов и реализуемых маршрутах обработки информации. Методы тестирования потоков данных обеспечивают выявление ошибок в вычислительной части программ и в процессах преобразования различной информации.

Совокупность спецификаций тестов может рассматриваться как ***второе, независимое описание содержания и реализации последовательных процедур комплекса программ***. Жизненный цикл и увеличение тестов при производстве и сопровождении комплексов программ должны проходить во времени, параллельно процессам и

динамике корректировки текстов программ. Тесты и сценарии их реализации должны быть адекватными и полностью ***отражать содержание текстов компонентов комплексов программ***, но в иной форме.

Процессы и методы тестирования программных модулей и компонентов

Относительная простота программных модулей (ПМ) позволяет детально анализировать их внутреннюю структуру и любой маршрут исполнения программы. Это обеспечивает возможность реализации ***двух стратегий тестирования***: от структуры и от данных. Этим двум стратегиям соответствуют два метода тестирования программ: метод анализа потоков управления и анализа потоков данных. При ***первой стратегии*** (рис. 2.5) за основу принимается структура ПМ, построенная по тексту программы в виде графа. В графе программы по некоторым критериям выделяются и упорядочиваются маршруты исполнения программы и условия-предикаты, при которых они могут быть реализованы.

Эти условия используются для подготовки тестовых наборов, каждый из которых должен реализоваться по маршруту, принятому за эталон при подготовке теста. Отклонение исполнения теста от первоначально выбранного маршрута рассматривается как ошибка, причина которой может быть либо в первичной структуре ПМ, либо в реализации конкретного маршрута при данном тесте на входе.

После устранения ошибок и несоответствия, выбранных и реализованных маршрутов, для каждого из них проверяется процесс обработки данных, и выявляются ошибки в результатах их преобразования. Затем оценивается достаточность выполненного тестирования ***по степени покрытия исходного графа программы*** проверенными маршрутами, которые выделялись по выбранному или заданному критерию. Завершается тестирование при требуемом покрытии графа программы протестированными маршрутами или при использовании ресурсов, выделенных на тестирование. В последнем случае необходима оценка достигнутой корректности программы и регистрация этой величины.

При этой стратегии некоторые выделяемые маршруты могут оказаться принципиально нереализуемыми из-за противоречивых условий в последовательных операторах – вершинах графа программы.

Стратегии выбора тестов для испытаний программных модулей и компонентов

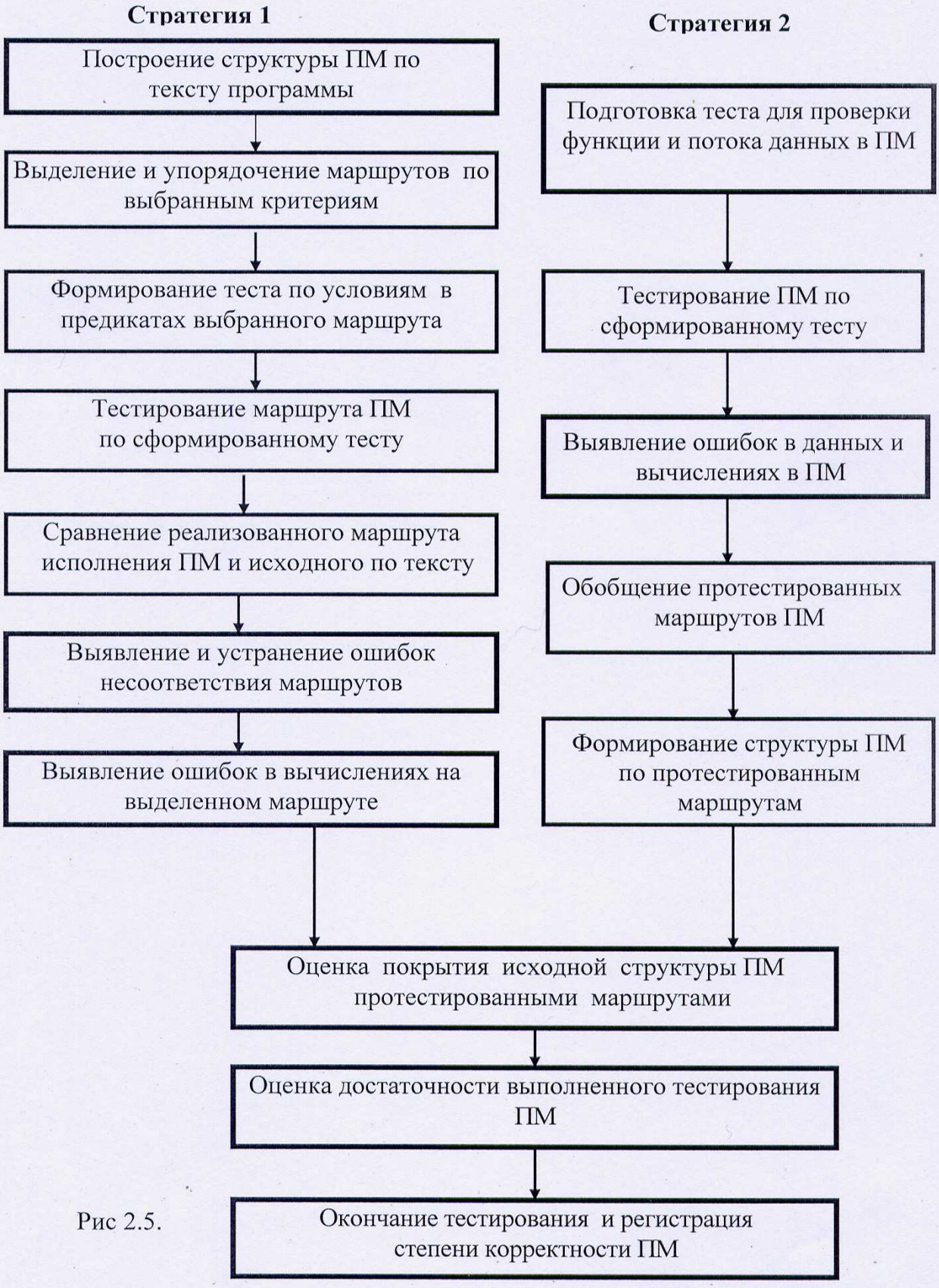

Рис 2.5.

Вследствие этого для некоторых модулей могут подготавливаться тесты, которые являются избыточными и не отражают реальное функционирование программы. Тем не менее, данная стратегия имеет преимущества при тестировании логических программ с малой долей вычислений.

При ***второй стратегии*** (см. рис. 2.5) за основу принимаются требования спецификаций, конкретные тестовые и эталонные значения, которые подготавливаются специалистами путем анализа переменных и предикатов в тексте программы. При каждом тесте программа исполняется по определенному маршруту, который регистрируется. При этом реализованный, в соответствии с анализируемыми требованиями, маршрут и поток данных рассматриваются как эталонные компоненты структуры программы. По мере тестирования отмечаются проверенные операторы, и оценивается полнота ***покрытия тестами требований спецификаций*** на маршрутах тестирования. Накопление и обобщение реализованных маршрутов позволяет ***воспроизвести структуру программы*** и контролировать степень покрытия каждого компонента. Для этого на каждом этапе тестирования выделяются компоненты программы, оставшиеся непроверенными, для которых необходимо подготавливать дополнительные тесты.

Данная стратегия имеет преимущества при сравнительно простой структуре программы и при преобладании в ней вычислительных операторов. На каждом маршруте, упорядоченное варьирование переменных акцентирует внимание на вычислительной части программы, что существенно для такого класса ПМ. Однако возрастает риск пропустить сочетания предикатов, определяющих непроверенный маршрут. Поэтому практически всегда целесообразно ***совместное применение двух стратегий***, с акцентом на одну из них в зависимости от особенностей тестируемого ПМ или его частей. Программы с преобладанием сложной логической структуры и с относительно малой вычислительной частью целесообразно начинать тестировать по первой стратегии и только для маршрутов с вычислительными операторами использовать анализ потоков данных (вторую стратегию).

Известно, что в крупных программных продуктах исчерпывающее тестирование, гарантирующее полное отсутствие в них дефектов и ошибок, ***принципиально не возможно*** и число дефектов, обнаруживаемых в программах даже независимыми тестировщиками. имеет пределы. На практике учитывать степень покрытия тестами доста-

точно трудоемко и зачастую желательно иметь более простой способ регламентирования и оценивания полноты тестирования. Возникает проблема, как практически учесть ограничения ресурсов и ***когда прекратить тестирование***, а также какая при этом будет достигнута ***корректность программ***, и способна ли она удовлетворить заказчика и пользователя при эксплуатации программного продукта.

Регулярная регистрация числа выявляемых дефектов за заданное время определенными тестировщиками, в результатах работы конкретных программистов, позволяет оценить усредненную ***интенсивность выявления дефектов при тестировании*** на предприятии при разработке определенных типов проектов. Такие оценки могут ***ориентировать*** при прогнозировании достигнутого качества соответствующих программных компонентов.

Системы тестирования и отладки программных компонентов высокого и контролируемого качества должны обеспечивать[2, 3, 44]:

- удобный, дружественный диалог в символьном и графическом видах, пользователей со средствами автоматизации тестирования;
- использование достаточно развитой и эффективной базы данных производства (репозитория) для накопления и хранения разнообразной информации о разрабатываемых программах, их версиях, планах тестирования, тестовых и эталонных данных, выполненных корректировках;
- автоматизированное планирование тестирования компонентов, выдачу рекомендаций пользователям по систематическому применению методов, стратегий и средств динамического тестирования отладки;
- оценку достигнутой корректности компонентов по выбранным критериям тестирования и определение основных показателей качества созданных программных компонентов;
- автоматизированную регистрацию и документирование всех выполняемых изменений в программах, и учет версий программных модулей и групп программ, в которых проведены корректировки.

Для отладки программных компонентов, входящих в сложные комплексы программ, и пригодных для повторного использования в различных проектах, необходим комплекс средств автоматизации, использующий основные современные методы выявления ошибок и дефектов в программах. Эти средства можно разделить на:

- ***статические*** – анализирующие спецификации и исходные тексты программ без их исполнения в объектном коде;
- ***динамические***, при использовании, которых программы функционируют в объектном коде и пригодны для их реального применения.

Эти средства объединяются средствами ***интерактивного диалога с пользователями и с базой данных производства***. В группу ***статической отладки*** входят средства, автоматизирующие структурный анализ и проверку формальной корректности текстов программ и выявление соответствующих типов ошибок. Средства подготовки данных для планирования тестирования должны обеспечивать выделение типовых структур в модуле (циклов, альтернатив, маршрутов исполнения) и их упорядочения по некоторым правилам. Выделение маршрутов исполнения программы и условий их формирования позволяет определить границы областей изменения переменных, влияющие на формирование тестовых наборов и их объем.

В группу ***средств статической отладки*** могут входить средства расчета длительностей исполнения модулей и компонентов. Эти средства позволяют получать ориентировочные значения и распределения длительностей счета программы аналитически, без ее исполнения на компьютере. В результате расчетов выявляются компоненты программы, требующие избыточно большого времени счета на реализующем компьютере, а также подготавливаются данные для общей ***проверки реализуемости программного продукта в реальном времени***.

Средства трансляции заданий с языка отладки обеспечивают обработку и подготовку к исполнению тестов и сценария отладочного задания. В задании указывается тестируемая программа, контролируемые и регистрируемые переменные, и состояния программы в процессе исполнения. Тестовые значения преобразуются в форму пригодную для исполнения отлаживаемой программой. Операторы отладочного задания объединяются с отлаживаемой программой или подготавливаются для исполнения в режиме эмуляции или интерпретации.

Средства исполнения программы по отладочному заданию управляют реализацией операторов задания в процессе исполнения отлаживаемой программы. В процессе обработки тестов производится предварительная селекция результатов тестирования в соответствии с указаниями операторов отладочного задания. В некоторых слу-

чаях получаемые результаты могут, автоматизировано сравниваться с эталонными значениями для выявления ошибок.

Средства регистрации и обобщения данных о результатах тестирования осуществляют преобразование зафиксированных данных функционирования отлаживаемой программы на язык отладки для диалогового взаимодействия с пользователем. Для сокращения избыточной информации производится редактирование и селекция результатов исполнения операторов отладочного задания. Отображение результатов тестирования в мнемонической и графической форме, может обеспечиваться унифицированными средствами интерактивного взаимодействия пользователей с компьютером.

Для каждого отлаженного программного компонента должно обеспечиваться ***хранение основных тестов и отладочных заданий***, с использованием которых проведено тестирование. Это позволяет поддерживать высокое качество компонентов и при необходимости вносить в них корректировки. При проведении таких изменений соответственно расширяется и модифицируется состав применявшихся тестов и заданий. Вместе с тестами целесообразно хранить результаты оценки полноты тестирования и достигнутой корректности программного компонента.

Документирование процессов тестирования программных компонентов следует проводить непосредственно при выполнении каждой из соответствующих работ с определенными целями тестирования. Должен быть описан и документирован, упорядочен и конкретизирован весь ***технологический процесс тестирования и частные результаты*** на каждом этапе ЖЦ:

- исходные данные: технические задания и спецификации требований; комплекс эталонных значений; критерии качества результатов тестирования; ограничения доступных ресурсов для реализации тестирования;
- совокупность выбранных методов, стратегий и сценариев тестирования при реализации этапов и процессов;
- план тестирования компонентов;
- методы и критерии оценки достигнутого качества программных компонентов;
- входные и результирующие данные тестирования;
- графики организации решения частных задач тестирования и необходимые для них ресурсы;

- распределение ответственности между специалистами по компонентам и видам проверок;
- протоколы результатов тестирования и обобщенные отчеты о достигнутом качестве программных компонентов.

[2, 3, 44] [9, 12, 23] [20, 33] [35, 44] […]

Глава 2.3

Тестирование потоков управления и потоков данных заказных программных модулей и компонентов

Стратегии выбора тестов потоков управления программных модулей и компонентов

Для производства сложных заказных программных продуктов высокого качества необходимо использовать тестированные и испытанные ***модули и компоненты гарантированного качества***. Такие модули должны предварительно пройти систематическую, упорядоченную верификацию и тестирование, обеспечивающие их полное покрытие тестами, контроль и аттестацию. Для этого должны применяться специальные методы на базе регламентированного анализа потоков управления и потоков данных, которые позволяют детально и достоверно ***тестировать небольшие модули и компоненты,*** и неэффективны при тестировании сложных комплексов программ. Они разрабатываются и тестируются наименее квалифицированными специалистами тестировщиками и требуют применения тщательно контролируемых методов, которые гарантируют высокое качество всех модулей, так как их дефекты и ошибки определяют ***качество всего заказного программного продукта***.

Программные модули (ПМ) являются наиболее массовыми компонентами в сложных комплексах программ и ***требуют для тестирования суммарно наибольших затрат*** при их производстве. Затраты на тестирование каждого модуля пропорциональны сложности, которая зависит от его структуры, числа узлов и объема необходимых вычислений. Суммарные затраты на тестирование каждого модуля можно оценить по значению сложности его графа. Тестирование ***потоков управления*** в структуре модуля программы должно быть начальным этапом, так как при некорректной структуре модуля возможны наиболее грубые искажения выходных результатов и даже отсутствие некоторых из них. Оно состоит в проверке корректности при обработке тестов последовательностей передач управления и форми-

рования требуемых маршрутов исполнения модулей комплекса программ – рис.2.6.

Тестирование потоков управления программных модулей включает:

- подготовку и построение модели графа узлов и связей потока управления программы модуля:
  - преобразование текста программы модуля в предложения и предикаты узлов и связей графа модуля;
  - формирование критериев выделения маршрутов для тестирования ПМ;
  - выбор стратегий упорядочения выделенных маршрутов тестирования ПМ;
  - оценивание сложности тестирования потоков управления программного модуля;
- выделение, анализ и обеспечение корректности циклов графа программы:
  - выделение в графе модуля циклов, расчет их сложности, автономное тестирование итераций циклической части маршрутов;
  - тестирование выделенных циклов по выбранным внутренним маршрутам и итерациям;
- выделение и тестирование ациклических маршрутов графа модуля программы в соответствие с требованиями;
  - расчет сложности и проверка корректности структуры ациклического части графа модуля для исключения структурных ошибок;
  - выделение и упорядоченье маршрутов тестирования модуля с циклами по сложности, длительности или вероятности исполнения;
  - тестирование выбранных маршрутов, при входных величинах, соответствующих исполнению выбранного пути:
- фиксирование результатов исполнения каждого теста на выбранном маршруте с циклами, устранение дефектов и ошибок модуля:
  - определение соответствия тестирования модуля с циклами предполагаемому результату, оценка качества модуля, обнаружение и исправление ошибок;
  - подтверждение и документирование полной корректности выбранных маршрутов потоков управления модуля с циклами и завершение его тестирования.

Рис. 2.6.

В качестве исходной информации ***при тестировании структуры программ модуля*** используется схема связей между операто-

рами текста программы. По этой схеме может выделяться полный набор маршрутов исполнения программы, подлежащих тестированию. Для каждого выделенного маршрута по тексту программы формируется набор условий, определяющих его реализацию и используемый при создании соответствующего теста. Детерминированное тестирование структуры программных модулей имеет целью проверку корректности выделенных маршрутов исполнения программ и обнаружение в основном логических ошибок формирования маршрутов. На практике при хаотическом тестировании и отсутствии упорядоченного анализа потоков управления некоторые маршруты в программе оказываются пропущенными. Поэтому ***первая задача***, которая должна решаться при тестировании структуры программ, – это получение информации о ***полной совокупности реальных маршрутов*** исполнения в каждой программе. Такое представление маршрутов позволяет упорядоченно контролировать достигнутую степень проверки маршрутов и предохраняет от случайного пропуска отдельных маршрутов. В результате значительно повышается достигаемое качество программных компонентов, и тестирование приобретает ***планируемый систематический характер*** [2, 3, 23].

Анализ критериев тестирования и выделение тестируемых маршрутов удобно проводить, используя графовые модели программ. При планировании тестирования структуры программ возникают, прежде всего, ***две задачи***: формирование критериев выделения маршрутов для тестирования ПМ и выбор стратегий упорядочения выделенных маршрутов. ***Критерии выделения маршрутов*** для тестирования соответствуют структурной сложности программных модулей:

- покрытия графа программы минимальным количеством маршрутов, охватывающих каждый узел или связь графа хотя бы один раз – f_1;
- выделения линейно-независимых маршрутов на базе понятия цикломатического числа, – каждый маршрут отличается от всех других хотя бы одной связью (или одним узлом) – f_2.

Число маршрутов, выделяемых по этим критериям, зависит от структуры программы. В реальных программах часть маршрутов может быть нереализуемой из-за противоречий в условиях, анализируемых на последовательных участках маршрута. Это может приводить к сокращению числа маршрутов по любому критерию. Невыполнение правил структурного построения программ, наоборот, может приводить к возрастанию числа маршрутов. К значительному возрастанию

числа маршрутов обычно приводят циклы в программах. Планировать тестирование можно по одному из критериев или, используя последовательно более жесткие критерии выделения маршрутов, при которых соответственно возрастает объем и сложность тестирования. Несмотря на перечисленные обстоятельства, приведенные критерии являются достаточно удобными для практических оценок сложности и полноты тестирования структуры программ.

Стратегии упорядочения маршрутов могут учитывать сложность маршрута и тестов для его проверки (количество узлов в графе, операторов, условных переходов и циклов в маршруте, частость его исполнения при рабочем функционировании ПМ, сложность получения соответствующих эталонных данных и другие факторы). В первую очередь целесообразно производить проверку основной группы маршрутов с экстремальными значениями используемого критерия в пределах ресурсов, выделенных для тестирования. При имеющихся ограничениях некоторая часть маршрутов может оказаться не проверенной и характеризует достигнутую ***корректность данного программного компонента*** по выбранному критерию. Упорядочение маршрутов при планировании тестирования в ПМ может базироваться на использовании в основном ***трех характеристик программных модулей (компонентов)***:

- числа узлов (или связей), операторов в выделенных маршрутах или расчетной длительности их реализации (стратегия 1);
- числа узлов – альтернатив или условных переходов, определяющих формирование каждого маршрута (стратегия 2);
- вероятности исполнения реализуемых маршрутов при реальном функционировании программы (стратегия 3).

Эти стратегии тестирования позволяют сосредоточить внимание разработчика на анализе наиболее важных элементов ПМ. При ***стратегии 1*** первичному тестированию подлежат маршруты, наиболее длинные по числу операторов (узлов в графе) и по времени исполнения. Им соответствуют обычно маршруты с наибольшим объемом вычислений и преобразований переменных. Эта стратегия целесообразна при планировании тестирования программ, имеющих вычислительный характер обработки данных при небольшом числе логических условий и маршрутов исполнения программ. Выбранные первичные маршруты не обязательно являются наиболее сложными по логике функционирования.

При ***стратегии 2*** приоритет отдается маршрутам, наиболее сложным по числу анализируемых условий – альтернатив. Такая стратегия предпочтительна при тестировании логических программ с небольшим объемом вычислений. При обеих стратегиях на завершающие этапы тестирования остаются простые по вычислениям или по логике исполнения маршруты, для которых необходимы относительно короткие тесты. Это соответствует традиционной стратегии многих тестировщиков программ подготавливать вначале тесты с возможно большим охватом элементов тестируемой программы. В типичной программе на долю 3% операторов зачастую приходится до 50% времени исполнения всей программы. В результате изучения профиля программы могут быть выявлены наиболее критические по длительности или самые частые маршруты исполнения ПМ. Использование профиля программ позволяет сконцентрировать усилия разработчиков ПМ на проверке наиболее активно функционирующих элементах и выделить слабо проверенные части программ, которые исполняются редко.

Планирование тестирования структуры программных модулей в значительной степени может быть автоматизировано. Задача автоматизированных систем ***планирования тестирования*** состоит в выделении маршрутов программ по одному или нескольким критериям с последующим упорядочением по заданной стратегии. В результате тестировщик программы автоматизировано информируется о составе маршрутов в программе для проведения ***упорядоченного тестирования***. Кроме того, если фиксировать маршруты, по которым уже проведено тестирование, то их можно автоматически исключать из информирования и выдавать на регистрацию только группу маршрутов, подлежащих первоочередной проверке. Эти же данные могут использоваться для автоматического расчета полноты проверки и для оценки достигнутой корректности программы по каждому из критериев выбора маршрутов.

Сложность тестирования потоков управления программных модулей

Для оценивания сложности тестирования программных модулей разработаны методы, позволяющие их количественно рассчитывать и контролировать при тестировании. ***Сложность программы модуля*** определяется числом взаимодействующих элементов – узлов, числом связей между узлами и сложностью их взаимодействия. При функ-

ционировании программы разнообразие ее поведения и разнообразие входных и результирующих данных в значительной степени определяются ***набором путей – маршрутов***, по которым исполняется программа. Экспериментально установлено, что сложность (трудоемкость) тестирования программного модуля зависит не столько от размера текста программы (числа операторов, узлов или связей графов), сколько от числа отдельных путей ее исполнения, существующих в программе. Все маршруты исполнения программы модуля и возможной обработки данных должны быть проверены при создании программы и тем самым определяют сложность ее полного тестирования [2, 3, 23].

Выше отмечалось, что маршруты исполнения программного модуля можно разделить на ***два вида***: маршруты принятия логических решений и преобразования логических переменных (потоки управления), и маршруты исполнения преимущественно вычислительной части программы и преобразования переменных (потоки данных). ***Первый вид маршрутов*** является результатом функционирования схем принятия решений и преобразования логических переменных. Для логических переменных отсутствует сильная корреляционная связь между соседними значениями, и каждое изменение переменной может определять разные области пространства результирующих значений на выходе программы. Такое преобразование переменных обеспечивается алгоритмами со сложной логической структурой, содержащей ряд проверок логических условий, циклов для поиска и селекции переменных, а также логические преобразования переменных. В результате в программе образуется множество узлов ξ_i маршрутов обработки исходных данных, которые определяют ***сложность структуры программы***. В ряде случаев подтверждена достаточно высокая адекватность использования ***потока управления*** и структурной сложности программ для ***оценки трудоемкости тестирования или вероятности*** не выявленных ошибок и затрат на разработку программных модулей в целом. Сложность тестирования потоков управления в программных модулях можно оценивать по числу маршрутов – M_k , необходимых для их проверки, или более по суммарному числу условий – ξ_k (узлов в графе), которые необходимо задать в тестах для прохождения при тестировании всех маршрутов определенной k-й программы:

$$\xi_k = \sum_{i=1}^{M_k} \xi_i \ , \qquad (1)$$

где ξ_i – число условий-предикатов, определяющих i-й маршрут.

Маршруты второго вида обычно логически короче, чем первого, и предназначены для преобразования величин, являющихся зачастую результатами измерения некоторых физических характеристик (***поток данных***). Такие переменные могут быть связаны условиями гладкости, т.е. условиями малых изменений производных этих переменных по времени или по другим параметрам. При оценке сложности вычислительных маршрутов программ необходим учет числа операндов, участвующих в вычислениях. Кроме того, исходные и результирующие данные при тестировании должны принимать несколько значений. Во всем диапазоне исходных переменных следует выбирать несколько характерных точек (предельные значения и несколько промежуточных), при которых проверяется программа. В особых точках значений и сочетаний переменных и в точках разрыва функции необходимо планировать дополнительные проверки. Таким образом, сложность проверки k-го программного модуля – $\square_k$ будет определяться числом маршрутов – M_k исполнения программы и числом обрабатываемых операндов – l_i на каждом i маршруте, умноженном на число значений – ν_{ij} для каждой исходной j-й величины на этом маршруте:

$$\square_k = \sum_{i=1}^{M_k} \xi_i \times \sum_{j=1}^{l_i} \nu_{ij} \ . \qquad (2)$$

Расчет показателя сложности тестирования программного модуля по такой схеме имеет значительную неопределенность из-за выбора числа значений переменных и констант – ν_{ij} (в основном, особых точек и величин) при варьировании исходных и промежуточных значений ***потока данных***. В то же время доля вычислительной части во многих сложных комплексах программ ***управления и обработки информации*** относительно невелика. Ее можно оценить из того, что ***ориентировочное число*** умножений и делений в таких программах в сумме составляет обычно 1 – 3%, а общее число арифметических операций не превышает 10%. Поэтому ниже большее внимание уде-

лено анализу структурного тестирования модулей управляющих комплексов программ.

Структурная сложность программного модуля может быть рассчитана по числу маршрутов в программе – M_k и сложности каждого i-го маршрута – ξ_i . Эти показатели в совокупности определяют минимальную сложность тестов – ξ_k для проверки структуры программного модуля (1), а, следовательно, трудоемкость его тестирования и вероятность пропуска ошибки в программе. Выделение маршрутов исполнения программы, минимально необходимых для ее проверки, и оценка структурной сложности может осуществляться по различным критериям. При этом формирование маршрутов зависит не только от структуры программы, но и от значений переменных, на различных этапах обработки. Затраты на тестирование каждого модуля ориентировочно прямо пропорциональны сложности, которая зависит от его структуры и объема вычислений. Суммарные затраты на тестирование модуля $\mathbf{C}_k$ пропорциональны значению его сложности и ориентировочно можно определить выражением:

$$\mathbf{C}_k = \mathbf{c} \sum_{i=1}^{M_k} \left(\sum_{j=1}^{l_i} \nu_{ij} + \xi_i \right). \qquad (3)$$

Значение множителя трудоемкости тестирования маршрута – **c** зависит от степени автоматизации процесса тестирования и генерации тестов. В высокоавтоматизированных системах тестирования сокращаются затраты ручного труда на подготовку и анализ тестовых данных, однако увеличиваются затраты на машинное время, необходимое для генерации тестов и для автоматической обработки результатов тестирования. В зависимости от этих факторов значения коэффициента – **c** могут различаться в несколько раз, и их следует экспериментально определять для каждой системы автоматизации тестирования.

Для определения суммарных затрат на тестирование модуля необходима оценка вероятности пропуска ошибки (дефекта) в модуле, т.е. достигаемого качества тестирования при принятом критерии выбора маршрутов исполнения программы и числа значений каждой переменной для выбранного маршрута. Обычно устанавливаются некоторые правила выделения маршрутов и варьирования переменных, гарантирующие достаточно глубокое тестирование модулей при доступных ограниченных ресурсах. Эти правила можно выбирать эмпи-

рически по опыту аналогичных разработок и стандартизировать для определенного предприятия или проекта. Такое выделение маршрутов трудно формализовать, и оно может представляться излишне трудоемким для оценки показателей сложности тестирования программ. Поэтому, зачастую, используются простые критерии выделения маршрутов, учитывающие только потоки управления и структурные характеристики программных модулей, а сложные вычислительные элементы программных модулей выделяются в подграфы и тестируются автономно.

Наличие циклов в программе способно резко увеличивать сложность (и трудоемкость) их тестирования. Полное тестирование должно охватывать проверку каждого маршрута в цикле при всех возможных итерациях цикла и при всех сочетаниях циклов с маршрутами ациклической части программы. При возрастании любого из сомножителей (числа независимых ациклических маршрутов, проходящих через цикл, числа внутренних маршрутов тела цикла или числа его итераций) пропорционально растет их произведение, а, следовательно, и сложность тестирования. Поэтому исчерпывающее тестирование реальных сложных программ с циклами иногда практически невозможно.

В качестве критериев при оценивании сложности тестирования структуры программы с циклами можно использовать два, выше описанных критерия: число маршрутов M, выделяемых по критериям f_1 или f_2 ; суммарную сложность тестирования ξ, т.е. суммарное число узлов ветвления на всех выделенных маршрутах, соответствующее числу условий, которое необходимо задать в тестах при выбранном критерии f_1 или f_2. В динамике реального исполнения простейшего цикла между его итерациями могут существовать зависимости, по крайней мере, трех видов:

- на разных итерациях цикла исполняются независимо все возможные маршруты внутреннего тела цикла;
- на всех итерациях цикла исполняется один и тот же маршрут тела цикла или некоторая определенная их последовательность;
- на разных итерациях цикла в силу наличия семантических связей исполняется подмножество реализуемых маршрутов тела цикла, зависящее от данных или от номера итерации.

При ***первой зависимости***, которая встречается редко, возникает необходимость в полном переборе всех внутренних маршрутов тела цикла в сочетании с определенным числом итераций. В этом случае

сложность тестирования цикла определяется сразу обоими параметрами: числом маршрутов тела цикла и числом итераций, и приближается к сложности исчерпывающего тестирования. При ***второй зависимости*** число маршрутов в теле цикла практически не влияет на сложность цикла. Определяющим становится количество итераций, необходимых для тестирования вычислений в теле цикла (например, с учетом требуемой точности). При ***третьей***, наиболее распространенной зависимости, определяющим при оценке сложности тестирования является не число итераций цикла, а число маршрутов тела цикла. Простейшие оценки сложности циклических структур целесообразно проводить в предположении, что порядок реализации маршрутов тела цикла не зависит от номера итерации. В этом случае при оценке сложности циклических структур конструктивным является подход с позиции выбора минимально необходимого числа проверок итераций циклов.

Для оценки ***сложности структурного тестирования*** логических программ с простейшими циклами целесообразно уточнять критерии проверки сложности f_1 и f_2 . По критерию f_1 ациклическая часть программы покрывается минимальным числом маршрутов, в которые входят маршруты, проходящие через цикл, размыкающие его и образующие минимальное покрытие тела цикла. Кроме того, к покрытию добавляется маршрут, содержащий замыкающую дугу цикла. По критерию f_2 ациклическая часть программы проверяется количеством тестов, равным цикломатической сложности ациклической части программы. При этом к каждому такому маршруту присоединяются все примыкающие к нему циклы. Проверка каждого цикла осуществляется одним маршрутом, содержащим столько итераций, какова цикломатическая сложность тела цикла, а тело цикла покрывается линейно независимыми маршрутами (см. рис. 2.6).

При таких оценках определяющими факторами являются полнота проверки тела цикла, условий его замыкания и размыкания. При этих критериях сложность цикла наиболее просто оценить, представив его эквивалентным ациклическим подграфом (подграфами). Например, при критерии f_2 , эквивалентным циклу с одной точкой входа и одной точкой выхода будет линейный подграф, содержащий последовательно соединенные все линейно независимые маршруты тела цикла, связывающие его точку входа с точкой выхода. Такой эквивалентный ациклический подграф добавляется к каждому маршруту в покрытии ациклической части графа по критерию f_2.

Эквивалентной циклу с одной точкой входа и одной точкой выхода при критерии f_1 будет совокупность внутренних маршрутов, соединяющих точки входа и выхода и покрывающих тело цикла по этому критерию, а также маршрут, представляющий собой одну итерацию цикла, состоящую из его замыкающей дуги и маршрута из покрытия тела цикла. Совокупность этих маршрутов, объединенная в ациклический граф с общей точкой входа, будет в этом случае эквивалентным графом для всего цикла. Однако в этом случае к каждому из маршрутов в покрытии ациклической части графа программы по критерию f_1 добавляется только один маршрут из части эквивалентного ациклического графа.

В качестве ***примера предположим***, что число маршрутов в ациклической части программы равно M_1. Тогда полное множество маршрутов состоит из полной совокупности всех маршрутов M_1 в ациклической части программы и группы маршрутов тела цикла $M_ц$, в которой к каждому маршруту M_1 присоединено 1.2. итерации (витка) цикла, причем на каждой итерации выполняется, по крайней мере, один из внутренних маршрутов $M_ц$ тела цикла. Для графа, имеющего один цикл, требующего исполнения пяти итераций (витков) с тремя внутренними маршрутами, а также содержащего 10 ациклических маршрутов, проходящих через цикл, суммарное число маршрутов для полного тестирования равно $M^* = (3 \times 5)\ 10 = 150$. Такое число тестов трудно реализовать практически и приходится из них выбирать небольшое доступное число тестов, допуская возможные дефекты в не полностью проверенной части программы.

Для обобщенных оценок необходимо в конкретных проектах выделение классов типовых структур сочетания циклических и ациклических частей программ модулей. При применении вложенных циклов со сложной структурой и с большим числом ветвлений в теле цикла сложность тестирования резко возрастает и велика вероятность сохранения в программе не обнаруженных дефектов и ошибок. Поэтому в процессе проектирования программных модулей необходимо в ***максимальной степени упрощать циклические компоненты*** в структуре, ***запрещать*** зацепляющиеся, вложенные и сложные структуры тела цикла, которые во многих случаях при тестировании определяют достигаемую корректность программных модулей.

При ***тестировании модулей в составе компонентов*** необходимо учитывать их собственную сложность и межмодульных связей по управлению и по информации. Каждый модуль, протестированный

автономно должен пройти дополнительное тестирование после включения в группу программ и в составе группы. Затраты на тестирование модулей в составе группы программ должны учитывать относительные суммарные затраты на тестирование влияния взаимодействия в группе всех входящих модулей с коэффициентом $d_k < 1$, зависящим от полноты и качества реализованной автономной проверки k-го модуля. Если модули автономно предварительно не тестировались (например, при нисходящем тестировании), то $d_k = 1$ и затраты на тестирование ***каждого модуля*** войдут в затраты при тестировании группы программ в ***полном объеме***. При тщательном автономном тестировании каждого модуля можно полагать $d_k \sim 0{,}1 – 0{,}01$, т.е. в группе модулей необходимая трудоемкость на его тестирование может составлять только несколько процентов. В результате затраты $\mathbf{C_{групп}}$ на тестирование H модулей с учетом трудоемкости тестирования каждого модуля (4) в составе группы программ могут составлять:

$$\mathbf{C}_{\text{групп}} = \sum_{k=1}^{H} d_k \, \mathbf{c} \sum_{i=1}^{M_k} \left(\sum_{j=1}^{l_i} \nu_{ij} + \xi_i \right) \qquad (4)$$

При анализе сложности тестирования межмодульных связей по управлению и по информации следует учитывать, что тестирование каждой информационной связи может быть необходимо при нескольких значениях каждой переменной, что соответственно увеличивает затраты. Оценки относительной сложности тестирования групп программ способствуют правильному распределению ресурсов (времени, труда специалистов и т.д.), выделяемых на тестирование групп разной сложности.

Корректность тестирования потоков управления программных модулей

Планирование тестов для модулей на базе потоков управления, их выполнения и устранения ошибок, целесообразно использовать при разработке технологий и Программ тестирования программных модулей и компонентов для обеспечения их соответствия требованиям высокого качества по следующим этапам (см. рис. 2.6):

- подготовка и построение модели графа узлов и связей модуля на основе текста программы;

- выделение, анализ и обеспечение корректности циклов, графа модуля программы;
- выделение и тестирование ациклических маршрутов графа модуля программы в соответствие с требованиями к качеству;
- фиксирование результатов исполнения каждого теста на выбранном маршруте, устранение дефектов и ошибок и завершение тестирования модуля (с циклами).

Эффективность тестирования обычно определяется полнотой проверки программного модуля или вероятностью наличия не выявленных ошибок в зависимости от затрат на создание тестов, исполнение программ и анализ данных тестирования. Эти затраты в значительной степени зависят от суммарной сложности тестов, проверяющих маршруты исполнения программы. На каждой дуге графа программы между узлами производятся вычисления и преобразования переменных, объем которых может изменяться в широких пределах. Для упрощения анализа тестирования структуры программ, можно предположить, что ***длительность и сложность вычислений на дугах графов программ одинакова и не велика***. Некоторые узлы графа программы могут образовываться в результате схождения дуг без последующего ветвления. Такие узлы не влияют на число маршрутов и их можно при анализе обобщать с ближайшим последующим узлом, в котором происходит ветвление. При этих предположениях сложность теста, проверяющего каждый i-й маршрут, в первом приближении пропорциональна числу дуг графа программы, входящих в этот маршрут, или числу условий ξ_i , которые необходимо задать в тесте.

Полная сложность тестов для проверки программы в последующем принимается равной сумме сложностей тестов ξ_i , используемых для проверки каждого i-го маршрута, где суммирование ведется по M_f маршрутам, выделяемым по одному из приведенных выше f-х критериев. Качество проведенного тестирования и ***достигнутая корректность модуля*** определяются возможностью получить при его реальном функционировании искаженные результаты. В маршрутах исполнения программы, содержащих участки, не проверенные тестированием, наиболее возможно искажение результатов из-за не выявленных ошибок. Предположим, что до выполнения тестирования, вероятность ошибки в j-й дуге графа программы, входящей в i-й маршрут, равна q_{ij} . Кроме того, вероятность ошибки в j-й дуге, не зависит от ошибок в остальных дугах графа программы, т.е. результат

вычисления, либо полностью используется на этой же дуге, либо является итогом исполнения программы. Тогда вероятность получения правильного результата на конкретном i -м маршруте:

$$\mathbf{p}_i = \prod_{j \in i} (1 - q_{ij}).$$

В реальных условиях конкретное исполнение программы происходит по одному из всех возможных M_f маршрутов. Выбор маршрута определяется обрабатываемыми данными, которые влияют на направление ветвления в узлах графа программы. Реализация каждой, j-й дуги i-го маршрута исполнения программы зависит от вероятности π_{ij} выбора этой дуги при предшествующем анализе в узле условия ветвления, вследствие чего вероятность реализации i-го маршрута

$$\pi_i = \prod_{j \in i} \pi_{ij} .$$

В результате вероятность отсутствия проявления ошибки при реальном исполнении программы определяется произведением вероятностей выбора соответствующих дуг и вероятностей правильного исполнения этих дуг:

$$\mathbf{P}_i = \prod_{j \in i} \pi_{ij} (1 - q_{ij}) .$$

Предположим, что правильность выполнения конкретного маршрута не зависит от предыдущих исполнений программы и равна $\mathbf{P}_i$. Тогда ***полная вероятность правильного функционирования программы*** при произвольных исходных данных определяется вероятностью выбора различных маршрутов и корректностью их исполнения (отсутствия в них ошибок):

$$\mathbf{P} = \sum_{i=1}^{M_f} \mathbf{P}_i = \sum_{i=1}^{M_f} \prod_{j \in i} \pi_{ij} (1 - q_{ij}) .$$

Эта величина может рассматриваться как ***показатель корректности программы*** по результатам тестирования ее структуры. Следовательно, вероятность проявления ошибки при случайных данных на входе: $\mathbf{Q} = 1 - \mathbf{P}$. Значения вероятностей исполнения маршрутов без ошибок зависят не только от вероятностей ветвления в каждой вершине и выбора дуг графа программы, но и от критериев выделения маршрутов. Полное число маршрутов M_f , выделяемых по каждому f-му критерию, может значительно изменяться в зависимости от критерия их выделения. При завершении тестирования программного

модуля по всем маршрутам, выделенным по выбранному критерию, модуль следует считать полностью корректным по данному критерию выделения маршрутов. Однако, если после тестирования оценить степень корректности программы при выделении маршрутов по более жесткому критерию, то модуль окажется протестированным не полностью и соответственно вероятность ошибки может быть на равна нулю на дополнительных маршрутах, выделяемых по этому дополнительному критерию.

Таким образом, набор маршрутов M_f , выделяемых по каждому f-му критерию, является полным в пределах использования этого критерия, и суммарная вероятность исполнения этих маршрутов должна быть равна единице. По каждому набору маршрутов M_f суммарные вероятности реализации маршрутов, подсчитанные по значениям π_{ij} , могут отличаться от единицы. Для оценки ***вероятности ошибки при выделении маршрутов*** по каждому критерию значения **P** следует делить на вероятность реализации суммы всех маршрутов по f-му критерию:

$$\mathbf{P} = \sum_{i=1}^{M_f} \prod_{j \in i} \pi_{ij} \, (1 - q_{ij}) \Big/ \left(\sum_{i=1}^{M_f} \prod_{j \in i} \pi_{ij} \right) .$$

Следует подчеркнуть, что в предыдущем выражении вероятность q_{ij} может стать равной нулю ***после полного*** тестирования соответствующей j-й дуги. Когда тестируется последний маршрут из выделенных по f-му критерию, предполагается, что входящие в него дуги, не будут содержать ошибок после их проверки. Поэтому после тестирования последнего из M_f маршрута вероятность **P** для каждого критерия становится равной единице.

Полное тестирование при некотором f-м критерии выделения маршрутов может соответствовать ненулевой вероятности обнаружения ошибки при выделении маршрутов по более жесткому (f+1)-му критерию. Разность этих вероятностей может использоваться для оценки целесообразности перехода на более жесткий критерий выделения маршрутов при ограниченном их числе. Формальный анализ рациональных пределов тестирования и выбора критериев для конкретных программ затруднен разнообразием структур и характеристик программных модулей, а также различием требований, предъявляемых к корректности программ.

Тестирование потоков данных программных модулей

В ***графах потока данных*** значения объектов связаны с обрабатывающими узлами, в которых эти значения вычисляются. При применении графов потока данных узел может быть использован для представления нескольких различных вычислений и, таким образом, может иметь несколько различных объектов, ассоциированных с исходящими связями. Обычная практика – размещение имен объектов на связях, исходящих из таких узлов. При таком использовании эти имена характеризуются значениями исходящих связей. Поскольку исходящие из одного узла связи являются входящими по отношению к следующему узлу, они отражаются значениями входящих связей. Узел выбора данных вычисляет специальную функцию значения входящей связи для использования в исходящей связи. ***Входящие связи*** узла данных должны быть помечены условием предиката, при котором выбирается данная входящая связь. Его значение выбирает объект данных, соответствующий входящей связи. Обрабатывающий узел с одной или более входящими связями формируется, по крайней мере, с одной исходящей связью. Входящие связи обозначаются именами данных. ***Исходящая связь*** обозначает вычисленную функцию этих объектов данных. Запоминающие узлы при последующей обработке должны уточняться, какое именно значение переменной используется.

Тестирование ***потоков данных*** можно разделить на два этапа – рис. 2.7. ***Первый этап*** тестирования состоит в анализе узлов обработки данных, определяющих значения предикатов в операторах выработки логических решений при взаимодействии элементов в модуле программ. Эти решения влияют на изменения маршрутов обработки информации, что сближает в этой части метод тестирования потоков данных с тестированием структуры модуля.

Второй этап – тестирование ***подграфа обработки данных*** на соответствие заданным требованиям. Оно состоит в проверке вычислений по аналитическим формулам, или определение численных значений результатов решения задач, в зависимости от числовых или логических значений исходных данных [2, 3, 23].

Если сделать отдельно модель потока данных и модель потока управления, каждая приведет к различным тестам, и при таком подходе не будет значительного перекрытия созданных тестов. Поэтому полезно помещать модель потока управления ***внутрь модели потока данных***. Повторяющиеся процедуры в модели, большей частью определяются потоками данных.

Тестирование потоков данных программного модуля включает:
- построение и анализ графовой модели для подготовки тестирования потока данных:
  - преобразование спецификации и текста программы модуля, чтобы каждой функции соответствовало одно предложение – подграф потока данных;
  - перечисление функций – подграфов и создание структуры узлов модуля, охваченных потоками данных;
  - формирование списка, в котором находятся промежуточные подграфы данных, зависящие от вычислений;
  - регистрация модели функций, связей и подграфов данных, которые отражают их обработку;
  - проверка полноты и корректности программной модели потока данных;
- планирования, реализации и исполнения тестов для выявления дефектов и ошибок потоков данных в модуле заказной программы:
  - выбор маршрутов тестирования с использованием порожденных подграфов данных;
  - выделение и регистрация ожидаемых результатов – эталонов тестирования выбранных маршрутов и подграфов данных;
  - выполнение тестов потоков данных на выбранных маршрутах и подграфах данных;
  - сравнение результатов тестирования маршрутов и подграфов данных с эталонами, регистрация дефектов и ошибок;
  - исправление дефектов и ошибок и регрессионное тестирование маршрутов и подграфов данных;
- тестирование модулей и компонентов заказных программ с учетом значений переменных и констант;
- базовые документы, регламентирующие результаты тестирования потоков управления и потоков данных модулей и компонентов, применяемых в заказных программных продуктах.

Рис.2.7.

Можно поместить модель потока данных ***внутрь модели потока управления***. Тогда обработка моделируется при помощи графа потока данных. Выбор способа совмещения моделей потоков зависит от того, какой из них является более сложным и доминирующим.

Данные, участвующие в вычислениях, должны быть определены явно в потоках данных по имени, типу и способу доступа. Это позво-

ляет рассматривать программу в виде мультиграфа, заданного структурой передач управления ***(потоком управления)*** и графом преобразования данных, участвующих в вычислениях ***(поток данных)***. Пересечение потоков управления и потоков данных осуществляется в узлах ветвления: проверки условий и циклов. Совместный анализ потоков управления и данных позволяет проверять корректность областей определения переменных на маршрутах исполнения программы. Последствия ошибок данных в модуле могут проявляться как малые изменения некоторых переменных в процессе вычислений, и как полное искажение или отсутствие на выходе требующихся величин. Тестирование программного модуля целесообразно проводить на упорядоченных наборах данных с учетом степени их влияния на выходные результаты. С этой позиции для последующего анализа целесообразно выделить ***два вида обработки данных***: способный полностью изменять область определения результатов и изменяющий результаты в пределах некоторой ограниченной области определения [2, 3].

Первому виду обработки соответствуют исходные данные в критических точках (узлах) и на границах областей изменения переменных. При таких критических значениях может ***изменяться маршрут*** исполнения программы, вследствие чего возможно наибольшее изменение результатов. Поэтому обычно тестирование обработки данных, прежде всего, направлено на проверку исполнения программ при значениях переменных, влияющих на выбор маршрута и логику функционирования программ (стратегия областей).

Граничные условия – это ситуации, возникающие в непосредственной близости к границам областей изменения обрабатываемых переменных. Число таких критических значений каждой переменной может быть на несколько порядков меньше, чем число значений по всей внутренней части области изменений этой величины. Большинство критических значений (предикатов) может существенно влиять на результаты и подлежит наиболее тщательному тестированию. В этой части тестирование обработки данных по содержанию близко к тестированию структуры (потока управления) программы. При этом виде тестирования маршруты формируются в процессе анализа и обработки данных на последовательных операторах условий в тексте программы. Сочетания исходных данных в тестах непосредственно влияют на степень охвата тестированием участков программы модуля. Путем сопоставления проверенных маршрутов с маршрутами, вы-

деленными по графу программы при различных критериях можно ***оценивать полноту тестирования потока управления модуля*** и приблизительную его корректность.

Второму виду обработки соответствуют данные в ограниченной или неограниченной области определения, которая может делиться на некоторое множество сопрягающихся областей (подобластей). Изменение данных в пределах такой области не влияет на маршрут исполнения программы. Поэтому для проверки функционирования программы из всего множества значений достаточно использовать при тестировании только несколько значений внутри и вблизи границ области. Количество величин, используемых для тестирования при обработке этого вида, может быть на несколько порядков меньше полного числа значений каждой переменной в области. В процессе такого тестирования проверяется точность осуществляемых вычислений, правильность размерности обрабатываемых величин, корректность формирования логических величин и т.д. При этом тестирование должно охватывать всю область изменения каждой обрабатываемой переменной и каждой результирующей величины.

Процессы формирования и тестирования модели потоков данных (без циклов) представлены на рис. 2.7. ***Тестирование с применением графовых моделей*** программных модулей способно обеспечивать высокое и управляемое качество модулей. Оно состоит из ***двух технологических этапов***: построения и анализа графовой модели для подготовки тестирования и этапа планирования, реализации и исполнения тестов для выявления дефектов и ошибок в программе. При планировании тестирования на основе объединенной графовой модели потоков управления и данных подготавливаются исходные данные требований и эталонов и обобщенный граф модели модуля. Эти данные используются для определения предикатов и переменных правильных маршрутов исполнения программы и описаний тестов. Маршруты формируются в соответствии с заданной стратегией и критерием выделения и упорядочивания маршрутов.

Подготовка тестов потоков данных модуля начинается с проверки корректности его спецификации, идентификации входных переменных и констант, присвоение каждой входной переменной имени и создание для нее входного узла. Текст программы модуля преобразуется так, чтобы каждой вычисляемой функции соответствовало одно предложение. Создается структура компонента, начиная с элементов, которые зависят от входных переменных и от результатов преды-

дущих функций, пока не охвачены все. Производится проверка промежуточных функций, является ли их упорядочение необходимым или просто удобным, можно ли упростить модель, удаляя промежуточные узлы или, добавляя промежуточные узлы и подграфы для сложных вычислений. Следует ***зарегистрировать модель потока данных*** – поименованный набор узлов – функций, их связей и подграфов данных, которые отражают их обработку.

Планирование и реализация тестирования модуля с использованием графов потоковых моделей программ начинается с формирования задания, эталонов и ограничений для планирования тестирования программного модуля (см. рис.2.7). На основе подготовленного графа программного компонента должно производиться выделение ***циклических подграфов*** для автономного тестирования, использование предикатов и маршрутов с циклами для формирования набора тестов и выполнение их тестирования. После этого целесообразно осуществлять выбор маршрутов тестирования по узлам и порожденным ими сложным вычислительным подграфам, и активизирование ***тестирования порожденных подграфов***. В результате возможно преобразование объединенной графовой модели к ациклическому виду. В этой модели производится выделение маршрутов по выбранному критерию и упорядочение выделенных маршрутов графа по выбранной стратегии ациклического графа исполнения программы. Далее целесообразно провести анализ и исключение нереализуемых маршрутов графа по противоречивым условиям в предикатах или на граничных значениях переменных. Оставшийся список предикатов и выделенных маршрутов используется для предсказания и записи ожидаемых итогов тестирования маршрутов, формирования набора тестов и реализации тестирования, по выбранным маршрутам и подграфам. Это позволяет оценивать полное число тестов использованных для тестирования выделенных маршрутов с учетом значений эталонов и ограниченных ресурсов, а также ***степень покрытия модуля тестами***, проконтролировать результаты тестирования, выделенные дефекты и ошибки. При этом должна быть проведена оценка качества программного модуля и допустимости его применения в проектируемом комплексе, ***подтверждена корректность значений результатов в промежуточных узлах, циклах и подграфах.***

Тестирование модулей программ с учетом значений переменных и констант

При анализе обработки данных в пределах внутренних областей их определения, методы тестирования целесообразно применять упорядоченно в ***следующей последовательности***:

- тестирование корректности ***записи и считывания переменных*** при вычислениях и полноты состава выходных данных на всех маршрутах исполнения программы;
- тестирование ***точности результатов вычислений*** и корректности обработки каждой переменной или константы;
- тестирование на полное ***соответствие состава и значений выходных данных*** требованиям программной спецификации.

В приведенной последовательности частные методы тестирования обработки данных позволяют, прежде всего, выявлять первичные ошибки, которые способны искажать результаты в наибольшей степени. При ограниченных ресурсах и такой последовательности тестирования в программе будут оставаться ошибки, наименее влияющие на корректность выходных данных. Полезно акцентировать внимание на выявление ошибок обработки данных, которые влияют на: логику исполнения программы, запись и считывание переменных, полноту состава результатов, точность расчета выходных данных и полное соответствие выходных данных требованиям спецификации модуля или компонента.

Предикаты, определяющие выбор маршрутов исполнения программных ***модулей заказных систем управления***, формируются в результате вычислений в узлах обработки на линейных участках программы. Эти участки в среднем невелики и содержат обычно около 10 операторов программы. Вычисления в большинстве случаев представляют собой простейшие линейные преобразования входных или промежуточных данных. По оценкам около 95% арифметических операторов включают только сложение и вычитание. Кроме того, предикаты обычно очень простые – в большинстве случаев с одной входной переменной и практически отсутствуют предикаты, использующие более двух входных переменных. Приведенные данные позволяют ограничить анализ линейными предикатами, характерными для широкого класса заказных программ управления.

Каждая ограниченная область исходных данных соответствует определенному маршруту в программе. ***Граница области*** определяется интерпретациями предикатов по маршруту и состоит из набора участков границы, каждый из которых определяется единственным простым предикатом, формирующим дугу маршрута в графе программы. Каждый участок границы области может быть открытым или закрытым в зависимости от оператора условий в предикате. Закрытый участок границы принадлежит ограничиваемой области и формируется предикатами с операторами $\leq$, $\geq$ или $=$. Открытый участок границы не входит в состав области и формируется операторами $<$, $>$ и $\neq$. Общее число предикатов в маршруте – это верхний предел числа граничных участков области входных переменных данного маршрута, так как некоторые предикаты маршрута могут в действительности не создавать граничных участков [2, 3].

Таким образом, программа по отношению к потоку данных, прежде всего, выполняет функцию ***разделения пространства исходных данных на области***, каждая из которых соответствует одному исполняемому маршруту. Ошибки в программе могут быть обусловлены модификацией границы области определенного маршрута, приводящей к расширению или сужению пространства исходных данных соответствующего маршрута. Кроме того, деформация границ областей может приводить к ошибкам уничтожения некоторых областей и потери соответствующих им маршрутов. Причинами таких ошибок могут быть искажения операторов анализа условий или искажения в процессе вычисления значений предикатов при правильном содержании оператора условия. В последнем случае обычно сдвигается граница области, однако она сохраняет общую структуру.

Один из достаточно часто встречающихся типов ошибок обусловлен искажениями условий формирования границ областей. Выбор тестовых данных вблизи границ областей обеспечивает наибольшую чувствительность к этим ошибкам. Тестовые исходные данные в зависимости от их положения относительно конкретной границы области можно разделить на два вида. Первый вид данных (***принадлежащая*** точка) размещается на границе области. При тестировании граничные точки входят в состав проверяемой области (условия $\leq$, $\geq$, $=$). Второй вид (***непринадлежащая*** точка) отстоит от границы на сколь угодно малую величину и находится на открытой стороне данной области (условия $<$,$>$, $\neq$). При тестировании такой области при-

надлежащие (граничные) точки относятся к смежной области, а в проверяемую область данная граница не входит.

Если маршруты исполнения программы ***соответствуют допустимым областям изменения входных данных***, то целесообразно проверить корректность основных операций обработки данных на выделенных маршрутах. Каждая величина на маршруте исполнения программы считывается из памяти и после использования для вычислений записывается в память компьютера для хранения и последующей обработки. Чередование операций чтения и записи переменных может быть нарушено в результате ошибок в программе. Для выявления таких ошибок проводится тестирование корректности записи и считывание реальных значений данных или статический анализ этих операций по тексту программы.

При ***тестировании потока данных*** выявляются все определения переменных, которые могут достигать каждой вершины графа программы по управлению. При этом под ***определением данных*** подразумеваются действия в программе, которые изменяют соответствующую величину данных. ***Использование данных*** – это операция в выражении, которая обращается к переменной, не изменяя ее. В результате для каждой точки программы устанавливается, какие ***переменные действуют*** в этой точке, т.е. какие данные, сформированные до прихода в эту точку, используются после выхода из нее. Анализ потока данных позволяет обнаруживать ошибки, обусловленные нарушением корректной последовательности ***операций определение – использование данных*** и выявлять переменные, которые целесообразно сохранять на регистрах.

Анализ процесса определение – использование, или ***«жизни данных»*** в программе естественно проводить по маршрутам ее исполнения. Для каждой единицы данных может быть установлена точка освобождения этой переменной, после которой она больше не используется и не определяется. Некоторые последовательности операций ***определение – использование*** над переменными могут быть аномальными. Так, например, последовательное изменение переменной без промежуточного использования на некотором маршруте, скорее всего, содержит ошибку в программе. Для выявления подобных аномалий необходимо упорядочить узлы графа программы так, чтобы операция в вершине не выполнялась до тех пор, пока не выполнятся все предшествующие ей узлы.

Тестирование корректности и точности результатов вычислений каждой переменной. Когда показано, что сочетания данных и их области определения соответствуют корректному формированию маршрутов в программе, а также нет явных ошибок в последовательностях определения и использования каждой переменной, целесообразно провести тестирование корректности обработки каждой переменной и точности вычислительной части программы. Этот вид тестирования производится преимущественно с вещественными и целыми величинами во ***внутренней части*** их областей определения.

Для каждой простой числовой переменной, кроме трех точек вблизи и на границе области определения, обычно необходимо тестирование программы в 3 – 4 промежуточных и в 2 – 5 особых точках значений входных данных. При 10 входных переменных и сложных вычислениях в программном модуле для тестирования вычислений может потребоваться до 50 тестовых значений. Группируя и упорядочивая тестовые значения разных переменных, их общее количество можно сократить до 5 – 10 тестовых наборов.

В вычислительных программах переменные часто представлены в виде массивов. При наличии массивов объем тестирования значительно увеличивается. Для проведения эффективного упорядоченного тестирования ***при переменных, образующих массив***, необходимо учитывать структуру и размер массива, а также наличие в структуре массива особых точек или подструктур. В простейшем случае (отсутствие особых точек в одномерной структуре массива и в значениях переменной) при тестировании необходимо минимум 3 значения переменной (краевые и промежуточные) при ее расположении в 3 точках массива, т.е. 9 тестов. При увеличении размерности массива и появлении особых точек в его структуре или значениях переменной соответственно возрастает число тестов, необходимых для проверки программы.

Отчет о результатах верификации и тестирования модулей и компонентов должен включать:

- справку о передаче модуля тетировщику на тестирование, когда в разработке участвуют независимые группы программистов и тестировщиков;
- журнал выполнения плана тестирования, используемый коллективом тестировщиков модулей и компонентов для регистрации: сценариев и операций во время выполнения тестирования; аномальных событий и дефектов для диагностики и локализации причин ано-

малий; изменений и корректировок, которые произведены в модулях и компонентах;

- результаты верификации и тестирования: результат выполнения (прошел/не прошел) для каждого анализа и выполненного теста и заключительный результат верификации и тестирования модуля и компонента; результаты оценивания покрытия и анализа трассируемости для выполнения тестов, и анализов в процессе верификации и тестирования;
- итоговый обобщенный отчет верификации и тестирования модулей и компонентов; результаты работ по верификации и тестированию, сценарии или виды проверки компонентов и комплекса в целом;
- удостоверение полученных данных тестирования модулей и компонентов в допустимых пределах отклонений от эталонных, заданных в требованиях на производство модулей и компонентов комплекса программ;
- удостоверение документации тестирования на модули и компоненты соответствует требованиям стандартов.

При тестировании модулей и компонентов комплексов программ основным ***лимитирующим ресурсом*** обычно являются допустимые трудозатраты специалистов, а также ограничения на сроки разработки версии программного комплекса. Ограничения реальных ресурсов на верификацию и тестирование модулей и компонентов определяют достижимое качество версий сложных заказных программных продуктов.

Глава 2.4

ПЛАНИРОВАНИЕ ПРОИЗВОДСТВА И ТЕСТИРОВАНИЯ ЗАКАЗНЫХ КОМПОНЕНТОВ И КОМПЛЕКСОВ ПРОГРАММ

Планирование производства компонентов и сложных комплексов программ

Производство модулей и компонентов составляет около половины затрат на полное проектирование и производство сложного заказного программного комплекса. Составление графика их производства, выполняемое менеджером проекта, необходимо для определения интегральных планов и ***характеристик процессов производства заказных программных комплексов***. Руководитель должен оценивать длительность этапов создания модулей, компонентов и всего проекта, определять виды и размер ресурсов, необходимых для реализации отдельных этапов и типов работ, и представлять их в виде согласованной последовательности. Если данный проект подобен ранее реализованному, то график производства компонентов нового продукта можно взять за основу. Если проект является инновационным, первоначальные оценки длительности и требуемых ресурсов для производства модулей и компонентов почти наверняка будут ***слишком оптимистичными***, даже если менеджер попытается предусмотреть все возможные неожиданности. С этой точки зрения производство программных комплексов не отличаются от больших инновационных технических проектов, в которых при планировании компонентов также неожиданно возникают проблемы и трудности.

Процесс планирования должен начинаться с определения ***проектных ограничений*** (временных ограничений, предельных возможностей и числа специалистов, бюджетных ограничений). Эти ограничения должны определяться параллельно с оцениванием проектных параметров, таких как размер и структура проекта, а также при распределении функций компонентов среди исполнителей. Затем определяются этапы разработки и то, какие результаты и документы (компоненты, подсистемы или версии программного комплекса) должны быть получены по окончании этих этапов. Далее на-

чинается циклическая часть планирования. Сначала разрабатывается план работ по выполнению проекта или дается разрешение на продолжение использования ранее созданного графика. По мере поступления новой информации о ходе выполнения проекта, возможен пересмотр первоначальных оценок плана и параметров всего проекта. Это, в свою очередь, может привести к изменению графика работ.

Цель состоит в том, чтобы подготовить ***реалистичную оценку затрат***, а затем обсудить соответствующие изменения плана и компонентов в общем графике производства комплекса программ. Строгое следование правилам декомпозиции компонентов и работ один из способов обеспечить полноту включения всей проектной группы, особенно опытных специалистов, в процесс оценки затрат. Для получения реалистичной оценки необходимо, чтобы она была основана на разумной продолжительности решения отдельных задач – рис. 2.8.

При организации структуры сложного комплекса программ, создаваемого большим коллективом специалистов, естественно возникает проблема оценки и упорядочивания модулей и компонентов ***целесообразных размеров*** на разных иерархических уровнях. Рациональные размеры программных модулей могут быть ограничены удобством и обозримостью разработки текстов программ и их тестирования отдельными специалистами. Хотя у некоторых ***программистов «виртуозов»*** есть тенденция писать монолитные модули размером во многие сотни строк, однако при этом возникают трудности тестирования и обеспечения гарантированной корректности таких программных компонентов. Экспериментально установлено, что во многих заказных программах управления и обработки информации реального времени – линейные участки программ между предикатами – узлами с ветвлением ***в среднем*** составляют около 5 – 10 строк. Поэтому число маршрутов исполнения программ и соответствующее число тестов, необходимых для их проверки, возрастает не пропорционально числу строк в программе, а значительно быстрее. Уже при ста строках в программе (10 – 20 предикатов – узлов ветвления) для ее тестирования может потребоваться более 100 тестов. Таким образом, при разработке модулей целесообразно учитывать рациональное ***ограничение их размеров на уровне трехсот строк текста***, что соответствует приблизительно тридцати альтернативам в таких программах [2, 15].

Планирование производства и тестирования компонентов и заказных комплексов программ включает:

- планирование производства компонентов и сложных комплексов программ:
  - планирование производства компонентов для комплекса программ;
  - выбор и документирование квалификации специалистов, необходимых для тестирования компонентов комплекса программ;
  - выбор, подготовка и утверждение детализированных планов производства и тестирования программных компонентов;
- подготовка и применение графиков производства и тестирования компонентов и комплексов программ:
  - освоение методики и средств планирования производства и тестирования компонентов сложных комплексов программ с использованием графиков Ганта;
  - принципы построения и применения сетевых графиков планирования производства и тестирования компонентов и комплексов программ;
- стратегии систематического тестирования сложных комплексов программ:
  - основные этапы систематического тестирования и испытаний сложного комплекса программ реального времени;
  - выбор, подготовка и утверждение детализированных планов тестирования программных компонентов;
  - анализ и контроль покрытия тестами требований и эталонов при тестировании компонентов и комплекса программ;
- Программа, график разработки и выполнения тестов для сложных комплексов программ:
  - подготовка Программы тестирования компонентов и комплекса программ;
  - подготовка графиков разработки и выполнения тестов для компонентов и комплекса программ;
  - формирование и контроль матриц соответствия требований и тестов компонентов и комплекса программ.

Рис. 2.8.

Программные компоненты высокой сложности целесообразно делить на более простые и легче тестируемые модули, что во многих случаях делается программистами интуитивно. Анализ числа тестов для большой реальной выборки программных модулей в заказных комплексах программ, создаваемых коллективом специалистов пока-

зал, что основная часть модулей содержала около 200 строк (до 20 – 30 узлов ветвления) [15]. Поэтому в ряде предприятий при разработке ***рекомендован рациональный размер программ модулей*** в пределах 100 – 300 строк текста, для полного тестирования которых достаточно использовать 10 – 50 тестов с суммарным числом условий ветвления около 100. При превышении рекомендуемых размеров модулей их трудно протестировать и ***целесообразно программистам делить на более мелкие компоненты***, доступные для практически полного покрытия тестами.

Каждый модуль должен рассматриваться с точки зрения ***ответственности за качество определенных личностей – программиста и тестировщика***. Коллектив или личность, выполняющие разработку, несут персональную ответственность за весь процесс и компоненты, даже если выполнение отдельных задач поручено другому предприятию. Принцип ***фиксирования ответственности*** лиц в архитектуре, компонентах и процессах производства программного комплекса облегчает применение стандартов и оценку качества характеристик частей конкретного проекта.

Группа специалистов должна начинать подготовку процесса производства с получения от менеджера или создания первичного ***плана производства модулей и компонентов***, а затем уточнять план. Для планирования процессов программирования и тестирования компонентов целесообразно формировать ***график поставки компонентов и модулей для сборки*** функциональных групп программ. При этом каждый компонент может быть вызывающим другие компоненты, и одновременно вызываемым из других компонентов и зависящим от результатов их функционирования. Эти связи определяют иерархическую схему взаимодействия между компонентами и модулями комплекса программ по управлению и по информации. Компоненты в сложном комплексе программ обычно наиболее сильно взаимосвязаны в пределах решения крупных функциональных задач. Поэтому целесообразно создавать ***два уровня графиков*** планирования программирования и тестирования компонентов сложного комплекса программ: на уровне взаимодействия крупных функциональных задач и на уровне взаимодействия компонентов и модулей в составе функциональных задач комплекса программ (см. главу 1.3).

План должен иметь реальные ограничения по числу и квалификации специалистов, по человеко-часам и по ***временному графику реализации*** каждого компонента. В плане также следует отражать

требуемое качество, предварительные условия и допустимые риски результатов производства компонентов. Сюда включаются все события, действия или обстоятельства, которые могут помешать созданию компонента требуемого качества в запланированный срок. При разработке плана должно быть учтено выделение функций, разработка которых имеет наибольшее значение для успеха проекта, и функций, разработка которых связана с наибольшим риском. Определение наивысшего риска дает возможность группе тестирования ***сосредоточить усилия на функциях и компонентах высокой значимости*** для пользователей и на достоверности результатов производства комплекса программ.

Менеджеры должны осуществлять текущий контроль за выполнением эффективной последовательности производства модулей, компонентов и комплекса программ, подготавливая как ***внутренние отчеты*** о развитии каждого процесса, так и ***внешние обобщенные отчеты*** для руководителя проекта и/или заказчика в соответствии с условиями договора. Все обнаруженные дефекты и результаты их устранения должны быть документально оформлены, а также в установленные сроки подтверждена полная реализация процессов и выполнение утвержденных планов производства. После создания всех запланированных программных компонентов и комплекса, менеджер должен определить степень их ***соответствия критериям*** качества, установленным в договоре.

Управление проектированием и производством программных продуктов регламентировано стандартом **ISO 16326**, где приводятся подробные рекомендации по планированию на основе требований стандарта **ГОСТ Р ИСО/МЭК 12207.2007** (см. главу 2.1). В стандартах детально изложены рекомендации по ***планированию и процедуры выполнения процессов управления*** на различных этапах производства комплекса программ. Выделенные для проектирования менеджеры должны отвечать за планирование и управление проектом, работами и задачами реализации планов производственных процессов, таких как заказ, поставка, разработка, эксплуатация, сопровождение или вспомогательные процессы. Подготовка и определение области планирования должны начинаться с установления требований к реализуемому процессу и продукту. Менеджер должен ***определить экономические возможности реализации*** планируемых процессов, проверяя наличие и достаточность ресурсов, выделенных для выполнения и управления процессами (специалистов, технологии и условий),

а также реальность сроков завершения производства программного продукта.

План должен иметь в своем составе ***детальные графики, идентифицирующие***: этапы работ; входные, выходные данные и описания решаемых задач; необходимые ресурсы и сроки выполнения; взаимосвязи и зависимости этапов и работ. Должно быть установлено организационно-техническое взаимодействие между различными группами специалистов, которые вносят свой вклад в процессы и обеспечение качества производства, а необходимая информация должна документироваться и регулярно анализироваться.

Исходные проектные данные и требования к планам производства комплекса программ, включая установленные законодательные и нормативные требования, должны быть оформлены документально, а их выбор проанализирован заказчиком на адекватность. Неполные, двусмысленные или противоречивые требования должны быть предметом урегулирования с лицами, ответственными за их предъявление. Спецификацию требований должен представить потребитель-заказчик. Однако по взаимному согласию ее может подготовить разработчик, в тесном сотрудничестве с потребителями для предупреждений разногласий путем, например, уточнения определений терминов, объяснения предпосылок и обоснования требований. Спецификация требований к планам может быть представлена и согласована в составе спецификации всей системы.

Выходные производственные данные при реализации планов должны быть документально оформлены и выражены так, чтобы их можно было проверить и подтвердить соответствие исходным проектным требованиям. Выходные проектные данные должны содержать ***критерии приемки продукта*** заказчиком или ссылки на них, а также идентифицировать те характеристики проекта, которые являются критическими для безопасного и надежного функционирования и применения программного продукта. К составу ***выходных проектных данных могут относиться***:

- описание результатов испытаний и применения программного продукта;
- комплект оформленного программного продукта и эксплуатационной документации для пользователей;
- комплект технологической документации для обеспечения возможности модификации и сопровождения версий программного продукта.

Планы могут ***эпизодически пересматриваться*** в процессе реализации производства. Для внесения изменений в план требуется специальная организация и регламентирующий документ, позволяющие отслеживать эти изменения и отображающие выполнение очередного этапа плана производства программного продукта. При планировании процесса производства программного продукта определяются контрольные отметки, регистрирующие окончание определенного этапа работ. Для каждой контрольной отметки создается краткий ***отчет***, который предоставляется руководству производством. Эти отчеты должны подводить итоги окончания отдельного, логически завершенного компонента и этапа плана производства.

Менеджер должен осуществлять текущий контроль за выполнением планов, подготавливая как внутренние отчеты о развитии каждого процесса, так и внешние обобщенные отчеты для заказчика в соответствии с условиями договора. После создания всех запланированных программных компонентов и продуктов, менеджер должен определить степень их ***соответствия критериям***, установленным в договоре. Для этого необходимо (см. рис.2.8) [7, 12, 35]:

- обеспечить координацию между планами разработки компонентов и функций для согласованного выполнения завершенных процессов жизненного цикла комплекса программ;
- планировать действия по разработке компонентов программного комплекса, и интегральные процессы жизненного цикла, которые позволят реализовать заданные требования и создать программный продукт требуемого качества;
- выбрать внешнюю среду поддержки и контроля жизненного цикла комплекса программ, включающую методы и инструментальные средства, которые целесообразно использовать для выполнения каждого процесса и обеспечивающие предотвращение дефектов и ошибок;
- выбрать и применять фрагменты стандартов, позволяющие обеспечивать требуемые качество, надежность и безопасность производства и применения программного продукта и системы;
- разработать документацию для планирования и последующей реализации возможных модификаций версий программного продукта.

Стандартами **ISO 16326** и **ISO 90003** рекомендуется в процессе планирования производства подготовить и утвердить содержание следующих ***основных детализирующих планов*** (см. рис.2.8):

- производства компонентов и комплекса программ, который должен определять используемую модель жизненного цикла комплекса и его компонентов;
- верификации и тестирования компонентов и комплекса программ, которые определяют методы и средства, способные удовлетворить последовательные цели процессов устранения дефектов и контроля качества программного продукта;
- обеспечения критериев качества компонентов и программного продукта, определяющего методы и средства, при помощи которых будет гарантировано их требуемое качество;
- реализации процессов интеграции компонентов в версии программного продукта;
- сопровождения и управления конфигурацией версиями программного продукта, которые должны устанавливать методы и средства, при помощи которых будут удовлетворяться цели процесса управления изменениями и корректировками комплекса программ;
- документирования результатов производства, выпуска технологической и эксплуатационной документации.

Кроме того, в составе перечисленных планов или автономно может быть полезной разработка ***вспомогательных планов***:

- тиражирования, адаптации и внедрения версий программного продукта для конкретных пользователей;
- обучения и подготовки пользователей для квалифицированной эксплуатации очередных версий программного продукта;
- обслуживания пользователей в процессе эксплуатации программного продукта.

Каждый представленный план должен учитывать ***экономику и ресурсы***, необходимые для его реализации, разделение работ на этапы и временной график выполнения этих этапов. Особого внимания требует ***сотрудничество разработчиков с заказчиком***, своевременное предоставление ему нужной информации и решение оперативных вопросов для обеспечения качества программного продукта. Целесообразно планировать и регулярно проводить разработчиком и заказчиком ***совместные анализы реализации планов проекта***, либо проводить такие анализы в случае значительных проектных событий, чтобы охватить:

- состояние и развитие выполняемых работ по производству и модификации программного комплекса и компонентов;

- соответствие результатов производства, согласованной спецификации требований заказчика;
- результаты проверок текущего состояния программного продукта и степени готовности к приемочным испытаниям заказчика;
- состояние подготовки и обучения конечных пользователей разрабатываемого программного продукта и системы.

Графики для планирования производства программных продуктов

Группа планирования должна составлять график разработки компонентов и комплекса программ, определить время и ресурсы, на эти работы. ***При подготовке графика*** должны назначаться специалисты, ответственные за выполнение каждой из работ по программированию определенных модулей, компонентов комплекса программ. Следует учитывать последовательность разработки текстов программ, их взаимозависимость и связь с созданием компонентов и функций комплекса программ. План должен ***определять, кто из специалистов отвечает*** за выполнение конкретного вида работ и компонентов. Также необходимо учитывать ***последовательность производства компонентов*** и их ***взаимозависимость***, поскольку при тестировании определенная функция зачастую не может быть выполнена, пока предыдущая функция не сформирует нужные данные. Необходимо планировать работы по адаптации внешней среды, подготовке обязательных отчетов в графике разработки и выполнения тестовых процедур. График тестирования должен учитывать проверку функций и компонентов, наиболее значимых для программного продукта, и функций повышенного риска.

В ***процессе составления графиков производства сложных программных продуктов*** весь комплекс работ разбивается на отдельные этапы и оценивается время и затраты ресурсов, требующееся для выполнения каждого этапа. Обычно многие этапы выполняются параллельно, график работ должен это предусматривать и распределять производственные ресурсы между ними оптимальным образом. Нехватка ресурсов для выполнения какого-либо критического этапа – частая причина задержки выполнения всего проекта. При расчете длительности этапов производства менеджер должен учитывать, что выполнение любого этапа обычно не обходится без больших или малых проблем и задержек. Разработчики могут допускать ошибки или задерживать свою работу, техника может выйти из строя, аппаратные

или программные средства поддержки процесса производства могут поступить с опозданием. Если проект инновационный и технически сложный, это становится дополнительным фактором появления непредвиденных проблем и увеличения длительности реализации проекта по сравнению с первоначальными оценками [7, 16, 35].

Кроме временных затрат, менеджер должен рассчитывать ***другие экономические и производственные ресурсы***, необходимые для успешного выполнения каждого этапа. Особый вид ресурсов – это ***«команда» специалистов***, привлеченная к выполнению проекта. Существует эмпирическое правило: оценивать временные затраты специалистов так, как будто ничего непредвиденного не может случиться, а затем значительно (на 30 – 50%) увеличивать эти оценки для учета возможных проблем. Трудно прогнозируемые проблемы существенно зависят от типа и параметров проекта, а также от квалификации и опыта членов команды разработчиков. Чаще всего для представления графиков работ используются ***диаграммы Ганта*** – рис. 2.9.

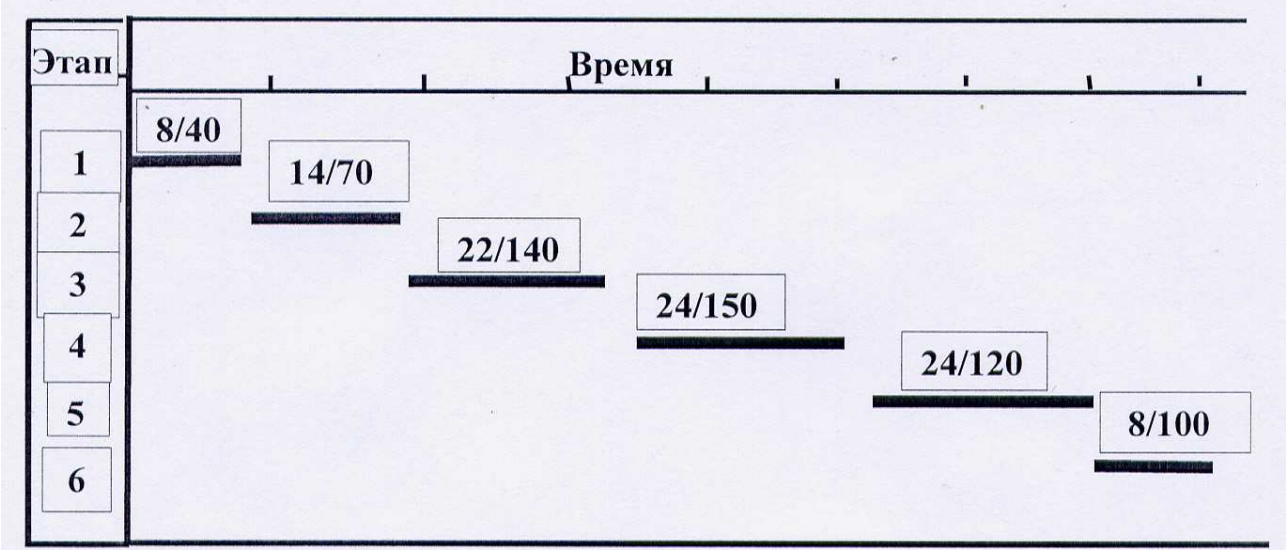

Рис. 2.9.

Графики Ганта ***– это методика планирования проектов***, которую можно использовать для достижения нескольких целей, включая календарное планирование производства, финансовое планирование, планирование использования специалистов и других ресурсов. График представляет собой гистограмму, где каждая горизонтальная линейка обозначает отдельный вид или компонент производственной деятельности. Линейки изображаются относительно временной шкалы планирования проекта. Длина каждой линейки пропорциональна продолжительности времени, запланированного на выполнение определенной

работы или этапа. График работ ***по всему проекту*** обычно представляется в виде набора диаграмм, показывающих разбиение производственных работ на основные этапы, зависимости между работами, распределение специалистов и затрат по этапам и работам (см. ***пример,*** на рис. 2.9 – шесть типовых этапов жизненного цикла, в числителе трудоемкость работ; в знаменателе – число специалистов).

Для его детализации может использоваться обобщенный перечень этапов и работ при проектировании и производстве сложных программных продуктов реального времени. Визуализация плана в виде ***графика Ганта*** может быть произведена по укрупненным этапам жизненного цикла и множеству частных работ и компонентов, составляющих эти этапы.

В составе и характеристиках работ и этапов можно отражать трудоемкость, длительность и число необходимых специалистов, а также их взаимодействие с предшествующими и последующими работами. ***Для каждого этапа***, в свою очередь, может быть составлен детализирующий его частный график Ганта, и оценены его ***экономические характеристики***. Необходимые временные, трудовые и другие ресурсы для выполнения и отражения работ на этих графиках, зависят от требований к детализации характеристик конкретного программного продукта и условий его производства.

Если итерационные методы предусматривают и допускают постепенное ***уточнение и расширение требований заказчика***, целесообразно назначать некоторую дату, после которой в проект не должны вноситься новые требования. Рекомендуется включать в график резервные периоды времени. Это делается из-за того, что невозможно учесть при планировании влияние всех внешних факторов. К тому же люди часто растягивают решение производственной задачи на все отведенное время. Если при этом в графике не предусмотрено буферного времени, то все временные интервалы наползают друг на друга и начинают перекрываться. Одним из вариантов предупреждения таких возможных неожиданностей является отведение под этапы заведомо большего количества времени. Другой вариант заключается в создании ***буферных интервалов в производстве***, во время которых не планируется решение каких-либо специальных задач.

Изменения, порожденные адаптацией такого рода, часто довольно существенны и дают результат, ***мало похожий на первичную структуру плана производства***. Структурирование должно использовать детальную информацию о проекте: правила конфигу-

рирования компонентов комплекса, наборы инструкций разработчикам, программистам и тестировщикам, рекомендации на формирование документации и программные интерфейсы компонентов. Кроме того, необходимо ***выделять ключевых специалистов***, для выполнения важнейших шагов производства, процесса интеграции компонентов, отслеживания дефектов или функциональных сборок, фиксации изменений, контроля стиля кодирования и взаимного просмотра кода, а также многие другие детали, специфичные для взаимодействия специалистов и компонентов любого сложного проекта.

График становится более детальным и достоверным по мере продвижения проекта и проведения ревизий. Как только утверждаются требования и архитектура комплекса программ, могут определяться частные производственные задачи и компоненты. В это же время могут более детально рассчитываться трудозатраты на каждый компонент и их взаимодействие, а также кто, когда и над чем будет работать. ***Пример фрагмента*** такого графика представлен на рис. 2.10.

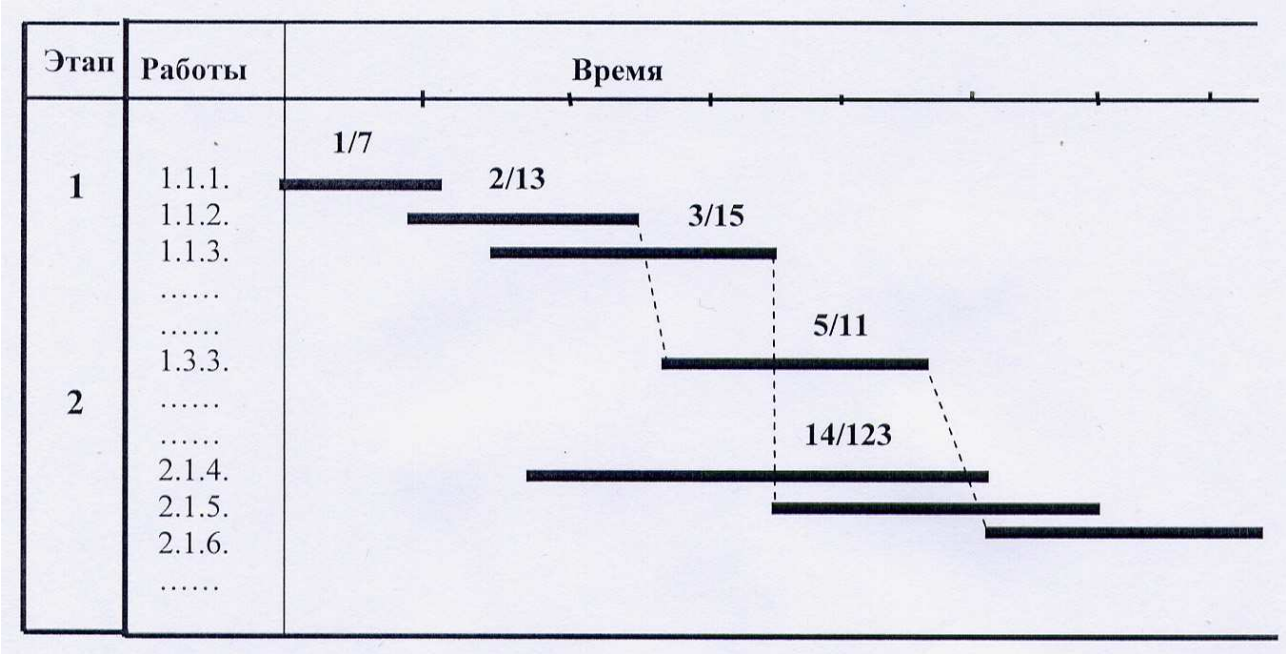

Рис. 2.10.

Слева перечислены производственные действия, идентификаторы компонентов (например, компонентов 71и 92) и их модулей (7.1.1 …9.2.6 и т.д.), а справа указаны горизонтальные полоски, длина которых соответствует длительности выполнения каждого этапа производства компонента в соответствии с временной шкалой. Трудоемкость процессов и число необходимых специалистов может указываться числами над линейкой для каждого модуля или компонента (рис. 2.10 – в числителе – число специалистов, в знаменателе

трудоемкость производства). Пунктирные линии отражают связи и зависимости начала некоторых работ только после завершения предшествующей работы. Например, начало работы 9.2.5 возможно только после завершения процесса 7.1.3.

Графики Ганта помогают выполнять ***календарное планирование производства***. Однако во время выполнения календарного планирования и его реализации могут быть определены новые виды деятельности, не предусмотренные во время первоначального укрупненного планирования. Тогда руководитель проекта должен вернуться назад и пересмотреть структуру декомпозиции работ и календарные планы для того, чтобы найти место и время вновь появившимся дополнительным видам деятельности.

При выполнении больших проектов, содержащих множество действий и отображаемой информации, диаграммы Ганта могут быть очень громоздкими. Они позволяют увидеть ***«общую картину» процессов производства***, однако ориентироваться в подобных графиках сложно. Поэтому полезно структурировать крупные графики на несколько более мелких по функциональным задачам или компонентам проекта, и выделять обобщающий сводный график производства всего программного комплекса без детализации процессов для компонентов (см. рис. 2.9).

График Ганта можно использовать для ***распределения ресурсов и планирования кадрового обеспечения***. Если необходимо спланировать работу и результаты конкретных специалистов по срокам, то можно воспользоваться графиком Ганта, на котором каждая полоса будет отражать работы одного из них. На таком графике ***специалисты являются ресурсами***, и на нем можно отражать их нагрузку во время производства. Можно пометить соответствующие части полос для обозначения количества времени, которое каждый специалист предположительно должен тратить на выполнение той или иной работы. Несмотря на то, что графики Ганта наглядно показывают каждый вид деятельности, они зачастую не отражают зависимостей между решаемыми задачами и компонентами.

Независимо от того, как ***выявляются отклонения от плана***, руководитель – менеджер должен принимать решение об их устранении. Там, где производство ограничивается доступным человеческим трудом, обычно коррекция производится добавлением специалистов или увеличением времени сверхурочной работы. Однако в программировании недостаток чаще всего ощущается не в рабочей силе как

таковой, а в ***интеллектуальных ресурсах конкретных специалистов***, поэтому с отклонениями от календарного плана здесь бороться сложнее. Формальное увеличение числа программистов или тестировщиков не поможет «уложиться» в поставленные сроки, настолько же важно учитывать, что помочь в этом могут интеллектуальные способности только некоторых особых специалистов. Таким образом, приемлемым вариантом может стать временный перевод подходящих специалистов на работу над проблемной частью проекта, либо наем специалистов – консультантов для устранения недостатков ресурсов.

Другой способ уложиться в заданные сроки – подробное, критичное ***изучение требований к программному продукту и исключение из них не самых актуальных***. Иногда, требования оказываются чересчур высокими, так как делаются попытки придать программному продукту дополнительные «украшения», практически не влияющие на его функции. Избавление от ненужных требований состоит в «***чистке» эталонов – требований***. Важнейшим принципом планирования производства комплексов программ является ***упорядочивание приоритетов*** в порядке возрастания важности требований для заказчика и/или пользователей. Зачастую можно согласовать с заказчиком и принять решение об исключении некоторых требований для того, чтобы уложиться в сроки.

Иногда после рассмотрения всех вариантов выхода из цейтнота, подходящее решение не найдено. Тогда следует ***признать неадекватности первоначальных планов и графиков***, и провести их пересмотр, исходя из нового понимания сложности поставленных задач, профессиональных способностей имеющихся специалистов и доступности ресурсов. Важно учитывать риски, связанные с прогнозированием графика, и быть готовыми к возможному пересмотру сроков. В менее тяжелых случаях принятие задержки результатов производства продукта может быть единственно правильным решением. Составляя, любой рабочий график, следует всегда учитывать ***принцип неопределенности***. Спланированный рабочий график – это предположение о будущих действиях разработчиков – программистов, тестировщиков и других специалистов, при оптимальных условиях. Возможными последствиями неопределенностей можно управлять с помощью предусмотренных в графике ***временных резервов***, которые являются аналогами экономических и финансовых резервов, применяемых при построении бизнес-планов.

Для того чтобы ***эффективно управлять неопределенностью***, целесообразно использовать в графике начальные оценки длительности производственных этапов. Некоторую неопределенность можно внести в буферы завершающего действия, что при непосредственной работе над проектом позволит управлять неопределенностью сразу для всех производственных действий. Вероятность своевременного завершения каждого этапа может составлять 50%, поэтому некоторые из них будут завершены позже установленного срока. Все разницы в сроках завершения этапов должны быть вычтены из буфера непредвиденных обстоятельств последнего этапа или всего производства в целом. Однако некоторые этапы могут быть завершены досрочно, поэтому размер суммарных сроков в буфере может оставаться практически неизменным.

Для планирования и мониторинга планов производства, создано несколько ***универсальных технологических программных продуктов***. Они обеспечивают представление планов в различной графической форме – ***графиков Ганта, сетевых и других графиков***. Большинство из них не имеют проблемного ориентирования и требуют перед применением подготовки и наполнения в них: состава, содержания и характеристик этапов, компонентов или работ, значений времени и трудоемкости, необходимых ресурсов и состава конкретных специалистов для реализации графиков. Такое итерационное планирование целесообразно проводить, последовательно детализируя содержание производственных процессов с учетом размера, функций и характеристик компонентов конкретного комплекса программ в виде ***совокупности графиков***:

- программирования и тестирования модулей и компонентов каждой выделенной функциональной задачи;
- каждой крупной функциональной задачи и взаимодействия ее модулей и компонентов в комплексе программ;
- этапов, производственных процессов и ресурсов всего жизненного цикла комплекса программ.

Суммарные затраты ресурсов, уточняемые при их оценках по этапам, компонентам и по перечням работ, ***не должны выходить за пределы***, ограниченные договором. Если эти условия не выполняются, то ***итерационно следует***: пересматривать и изменять или ресурсы, или графики и распределение ресурсов по этапам и компонентам производства, или сокращать состав и содержание работ на производственных этапах компонентов. Во всех случаях следует контролиро-

вать реализацию утвержденных требований к функциям, характеристикам и экономике программного продукта.

На основе значений длительности программирования и тестирования модулей и компонентов и взаимосвязи между ними может строиться ***сетевой график*** декомпозиции и последовательности их реализации. На этом графике должно быть видно, какие компоненты могут программироваться и тестироваться параллельно, а какие взаимозависимы и должны выполняться последовательно друг за другом. Тестирование ряда определенных компонентов не может начаться, пока не выполнены все компоненты на всех путях, ведущих от начала тестирования компонентов комплекса к данному компоненту.

Минимальное время выполнения всего производства комплекса можно рассчитать, просуммировав в сетевой диаграмме длительности производства компонентов ***на самом длинном маршруте*** от начала производства группы компонентов до его окончания (это так называемый ***критический путь)***. Таким образом, общая продолжительность производства компонентов проекта зависит от компонентов, находящихся на ***критическом пути.*** Любая задержка в завершении производства любого компонента на критическом пути приведет к задержке всего проекта. Задержка в завершении производства компонентов, не входящих в критический путь, не влияет на продолжительность создания всего комплекса программ до тех пор, пока суммарная длительность производства этих компонентов (с учетом задержек на каком-нибудь пути) не превысит продолжительности работ на критическом маршруте. [20, 31].

Стратегии систематического тестирования сложных комплексов программ

Стратегия тестирования – это совокупность методов и решений, вытекающих из целей и задач производства проекта и его тестирования, общие правила и принципы, способствующие достижению целей разработки программного продукта высокого качества. Ведущий тестировщик или тест-менеджер, готовящий план, должен тщательно обдумывать идеи, выводы и решения таким образом, чтобы специалисты, использующие план, четко понимали, что от них требуется: стратегию тестирования, планирование и использование ресурсов, управление конфигурацией проекта и минимизацию рисков – рис. 2.8.

При сравнении результатов функционирования проверяемых программ на соответствие эталонам используются ***критерии оценки допустимых отклонений от эталонов и решения о степени корректности*** комплекса программ. Величина допусков зависит от типа проверяемого алгоритма, метода и этапа проверки корректности программ. Для простых логических алгоритмов, реализующих некоторые схемы принятия решений, при анализе результатов тестирования обычно используется детерминированный критерий абсолютной идентичности проверяемых и эталонных решений при одинаковых исходных данных.

Интеграционное тестирование составляет объединение программного кода, соответствующего двум или большему количеству программных модулей, и тестирование полученного в результате кода компонента. Это должно гарантировать, что вместе они работают, как требуется, до полной интеграции и тестирования кода каждого функционального компонента. Так как отдельные модули могут включать другие модули, некоторая часть интеграции и тестирования модулей, может происходить в процессе модульного тестирования. Тестовые варианты должны покрывать все требования проекта уровня функциональных компонентов.

Тестирование функциональных компонентов в составе программных комплексов в процессе разработки комплексов программ и оценки полноты тестирования осуществляются преимущественно по степени выполнения требуемых функций и по характеристикам достигаемой корректности и качества функционирования КП в целом. Значительную помощь в повышении качества сложных, критических КП, может оказать систематизация видов тестирования и упорядоченное их проведение при разработке. Эти виды тестирования должны быть ориентированы на ***дифференцированное*** выявление определенных классов дефектов. Для каждого вида тестирования целесообразно разрабатывать методику его выполнения с указанием проверяемых компонентов, контролируемых параметров, ожидаемых и эталонных результатов.

Тестирование полноты решения функциональных задач при типовых исходных данных предназначено для обнаружения дефектов функционирования в нормальных, штатных условиях, определенных требованиями технического задания на базовую версию КП. Первичным эталоном являются цели и задачи создания программного продукта. Тестирование функционирования программ ***в критических***

ситуациях по условиям и логике решения задач (***стрессовое тестирование отказоустойчивости***) проводится при исполнении программ в нештатных ситуациях, которые редко реализуются, но важны для обеспечения качества и надежности функционирования системы обработки информации. Для разработки таких тестов создаются сценарии критических сочетаний значений исходных данных и условий решения задач, при которых необходимо проверить функционирование программ и можно ожидать искажения результатов и отказы.

Основные этапы систематического тестирования и испытаний сложного заказного комплекса программ реального времени и его компонентов представлены на рис. 2.11. При тестировании и испытаниях ***корректности функциональных компонентов комплексов программ выделены этапы***:

- комплексирование модулей и тестирование автономных функциональных групп программ в статике без взаимодействия с другими компонентами и возможно без подключения к операционной системе реального времени;
- тестирование функциональных групп программ в статике с учетом взаимодействия с некоторыми другими важнейшими компонентами и с базой данных;
- тестирование отдельных программных компонентов в реальном времени во взаимодействии с другими функциональными компонентами и с основными компонентами операционной системы и базы данных.

Сложность тестирования компонентов на этих этапах в значительной степени обусловлена ***несинхронным процессом производства и тестирования*** компонентов отдельными специалистами в коллективе. Первично спланированная логика сопряжения между собой отдельных компонентов и подключения их к операционной системе не всегда выполняется из-за задержек в разработке и автономной отладке некоторых из них. Целесообразная последовательность тестирования определенных компонентов может нарушаться неготовностью к сопряжению с ними других взаимодействующих программ.

Для обеспечения имитации объектов внешней среды и других взаимодействующих групп программ на этих этапах используются преимущественно детерминированные контрольные задачи или частные генераторы соответствующих тестов.

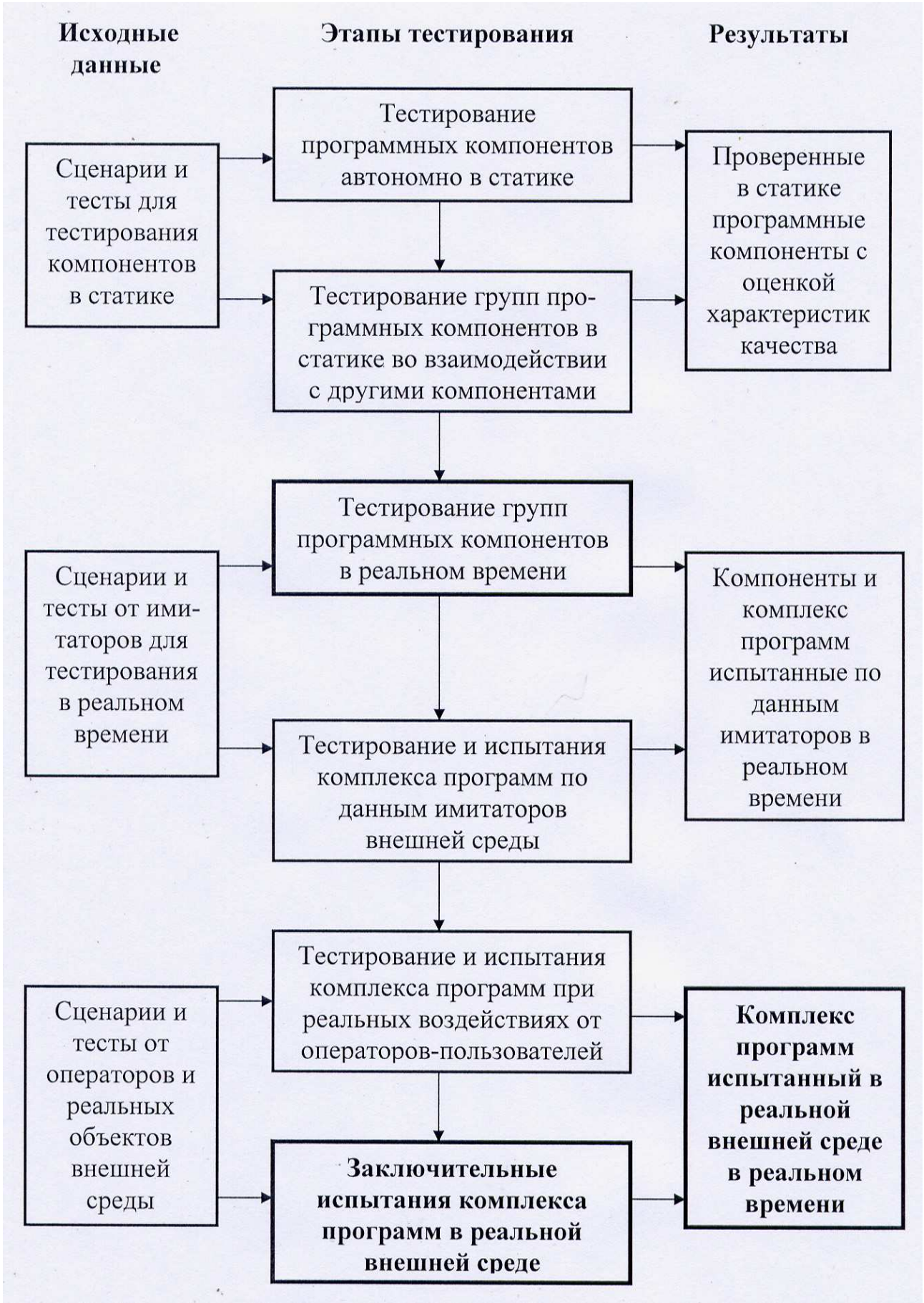
Исходные данные
Этапы тестирования
Результаты
Сценарии и тесты для тестирования компонентов в статике
Тестирование программных компонентов автономно в статике
Тестирование групп программных компонентов в статике во взаимодействии с другими компонентами
Проверенные в статике программные компоненты с оценкой характеристик качества
Тестирование групп программных компонентов в реальном времени
Сценарии и тесты от имитаторов для тестирования в реальном времени
Тестирование и испытания комплекса программ по данным имитаторов внешней среды
Компоненты и комплекс программ испытанные по данным имитаторов в реальном времени
Тестирование и испытания комплекса программ при реальных воздействиях от операторов-пользователей
Сценарии и тесты от операторов и реальных объектов внешней среды
Заключительные испытания комплекса программ в реальной внешней среде
Комплекс программ испытанный в реальной внешней среде в реальном времени

Рис. 2.11.

Эти генераторы тестов целесообразно разрабатывать и оформлять как отдельные модули или группы программ, функционирующие на той же технологического компьютера и в той же операционной среде, что и отлаживаемые компоненты. Совместно с ними также реализуются и функционируют частные специализированные программы для обработки и обобщения отдельных результатов тестирования соответствующих групп программ.

После комплексирования основных, функциональных компонентов начинаются их тестирование и испытания в составе комплекса программ. Для них наиболее характерны следующие ***этапы квалификационного тестирования и испытаний КП в реальном времени*** (см. рис. 2.11):

- по данным моделирующего стенда или генераторов тестов, имитирующих отдельные объекты внешней среды;
- с имитаторами отдельных объектов внешней среды и с реальными воздействиями от операторов-пользователей;
- в полностью адекватной реальной или имитированной внешней среде и с реальными воздействиями от операторов-пользователей.

На всех этапах тестирования, кроме операций непосредственной проверки функционирования ***заказных программ***, можно выделить еще две важных группы работ. ***Первая группа*** – это работы по методическому обеспечению процессов тестирования и по созданию средств автоматизированной генерации тестов. ***Вторая группа*** работ должна обеспечивать возможность обработки результатов тестирования и корректной оценки достигнутых характеристик качества функционирования программ.

Средства генерации тестов и обработки результатов тестирования можно разделить на ***три вида***. Одни и те же средства автоматизации тестирования в статике обычно обеспечивают отладку групп программ как автономно, так и во взаимодействии с другими компонентами. Средства, имитирующие внешнюю среду в реальном времени, чаще всего ориентированы на тестирование, как функциональных компонентов, так и комплекса в целом. Еще один вид генераторов тестов в той или иной степени использует реальные объекты внешней среды. Первоначально такими объектами являются имитирующие стенды с участием реального функционирования операторов-пользователей. Затем источниками тестов могут быть комплексы

реальной аппаратуры внешних объектов или их аппаратурные аналоги.

Планирование работ по тестированию должно учитывать ресурсы и работы, которые необходимо выполнить, чтобы своевременно ***подготовить тестовую среду***. Тестировщики должны определить требования к аппаратному, программному и сетевому обеспечению с целью создания и поддержки адекватных изменений тестовой среды. Нужно спланировать работы по приобретению, установке и настройке компонентов, моделей или динамических генераторов тестовой среды. Создание плана тестирования – ***итеративный процесс***, требующий обратной связи с различными участниками проекта и их согласия с определенными в нем процессами, стратегиями тестирования и сроками выполнения работ. Заказчик должен ***утвердить стратегию тестирования*** и тестовые процедуры, которые должны быть подробно описаны в плане тестирования, и определять какие сценарии и тесты когда будут выполняться. Кроме того, предполагается, что заказчик согласен с тем, что план тестирования и связанные с ним тестовые сценарии достаточно проверяют покрытие тестами системных требований и сценариев использования системы.

План тестирования должен определять объем работ по тестированию [3, 9, 29]. Обычно выстраивают ***структуру работ***, в которой на одном уровне определяются категории работ по тестированию, а на другом уровне – подробные описания работ. Структура детализации работ используется в сочетании с хронометражем для определения времени выполнения каждого из этапов тестирования. Кроме того, план тестирования должен отражать ***оценки затрат*** на тестирование. Оценка затрат может определять число сотрудников группы тестирования в проекте в часах или в количестве людей, если на выполнение определенного объема работ выделяется конкретный срок. По возможности в план тестирования помещаются такие оценки затрат, как планируемое число тестовых процедур и сценариев. Должны быть документированы ***квалификация и навыки специалистов***, необходимых для проведения тестирования. Тест-менедер должен оценить разницу между требуемой квалификацией и реальной подготовкой персонала, чтобы определить возможные ***направления обучения***.

Программа, график разработки и выполнения тестов для сложных заказных комплексов программ

К ключевым элементам планирования тестирования относится ***тестирование эксплуатационной и технологической документации***. Успешная и рентабельная Программа тестирования ***требует ясного видения целей*** и четкого понимания различных параметров Программы тестирования. Подробное ***изучение системных требований*** или сценариев использования системы вместе с тщательным определением параметров Программы тестирования и требований к тестам необходимы для эффективного тестирования программного продукта. Такая ***Программа тестирования***, включающая в себя средства автоматизации тестирования программного продукта, должна иметь свой ***собственный жизненный цикл***, в который входят планирование стратегий и целей, определение требований к тестам, анализ, проектирование и кодирование тестов. По аналогии с процессом, которому следуют при производстве комплекса программ, необходимо определить ***жизненный цикл тестов*** прежде, чем они будут спроектированы. Требования к тестам должны быть ясно сформулированы и документированы, чтобы все участники производства понимали, на чем основаны работы по тестированию.

Разработка тестов включает создание тестировщиками, сопровождаемых, многократно применяемых, простых и надежных тестовых процедур, что может потребовать не меньше усилий, чем разработка программистами тестируемых текстов программ. Чтобы добиться максимального эффекта от автоматизации тестирования, тестировщики должны вести разработку тестов параллельно с созданием программистами текстов программного продукта. Схема структуры комплекса программ является графическим представлением основных работ, которые должны быть выполнены во время разработки тестов. Группе тестирования требуется проводить корректировку и уточнение структуры разработки тестов, чтобы отразить приоритеты конкретного программного продукта.

Анализ полноты покрытия тестами требований при исполнении комплекса программ должен определять, какие требования не были протестированы и какие части структуры программного продукта не были исполнены при тестировании. Тестирование структурного покрытия должно устанавливать, не пропущены ли элементы

структуры программы, которые не проверены тестовыми процедурами и покрыли ли тесты всю структуру комплекса программ. ***Тестовые варианты, основанные на требованиях***, могут не полностью покрыть структуру программы. Анализ покрытия версии программного продукта тестовыми данными, основанными на требованиях, должен определить, насколько полно тестирование проверило реализацию всех требований, и показать потребность в дополнительных тестовых сценариях. Поэтому должен выполняться анализ полноты структурного покрытия и проводиться его верификация. [3, 9, 12].

При подготовке тестирования на основе требований к программному продукту, анализе технического задания и совокупности требований к функциям и характеристикам целесообразно оценивать, ***сколько нужно тестов для полного испытания версии программного продукта***. Если составлять тесты, руководствуясь этим принципом, можно оценивать, какое время потребуется на разработку, исполнение и анализ полного комплекта тестов при создании очередной версии программного продукта. Группа тестирования должна составлять ***план-график разработки и выполнения тестовых процедур*** по шкале времени реализации проекта, как средство определения временных рамок для разработки и выполнения различных тестов (графики Ганта см. выше). План-график определяет зависимости между тестами и общие сценарии, которые будут постоянно использоваться при тестировании. Перед созданием полного набора тестовых процедур для очередной версии программного продукта группа тестирования должна провести ***анализ связей между компонентами***. Результаты этого анализа помогут выявить независимые компоненты, спланировать зависимости между тестами и выделить общие сценарии, которые могут повторно применяться в процессе тестирования последующих версий. Для этого строится матрица связей, которая показывает взаимозависимость различных тестовых сценариев.

В ходе разработки тестовых процедур группа тестирования должна проводить ***конфигурационное управление и контроль*** для тестовых сценариев и тестовых данных, а также для каждой отдельной тестовой процедуры. Группы тестовых процедур и сценариев многократного применения обычно хранятся в списке или библиотеке тестовых данных – в тестовом архиве. Необходимо создавать базовые версии тестового архива при помощи инструментов конфигурационного управления (см. главу 2.6). Группа тестирования должна состав-

лять график разработки и выполнения тестовых процедур, чтобы определить время и ресурсы, на эти работы. ***При подготовке графика*** [10]:

- назначаются специалисты, ответственные за выполнение каждой из работ по тестированию;
- следует учитывать последовательность разработки тестов, их взаимозависимость и связь с процессами и компонентами ЖЦ комплекса программ;
- сокращаются или исключаются возможные конфликты между тестовыми процедурами, которые должны быть документированы;
- тестовые процедуры могут быть объединены в группы в соответствии с функциями программного продукта;
- разработка тестовых процедур и их применение должны быть спланированы таким образом, чтобы исключить дублирование работ;
- в структуре тестовых процедур следует учитывать и документировать их приоритеты и риски;

План-график должен помогать ***определить, кто из специалистов отвечает*** за выполнение конкретного вида работ. Составив детальный план-график разработки и выполнения тестовых процедур, группа тестирования сможет избежать дублирования работ. ***Необходимо описать установку и настройку тестовых процедур***, чтобы убедиться, что проведение этих работ соответствует стандартам управления конфигурацией**,** важно, чтобы тестирование проводилось в управляемой среде. Также необходимо описать ***последовательность*** выполнения процедур и их ***взаимозависимость***, поскольку при тестировании определенная функция не может быть выполнена, пока предыдущая функция не сформирует нужные данные. График разработки и выполнения тестовых процедур должен включать в себя тестирование, проверяющее различные циклы обработки, относящиеся к комплексу программ. Необходимо спланировать работы по настройке среды, тестированию подготовки обязательных отчетов в график разработки и выполнения тестовых процедур.

При составлении графика производства и выполнения тестовых процедур необходимо учитывать ***приоритеты*** и ***риски***, присвоенные различным тестам. Такие тесты должны выполняться в первую очередь, и график обязан предусматривать достаточно времени для проверки этих функций и, в случае необходимости, для регрессион-

ного тестирования. Группа тестирования должна отслеживать изменения требований и тестов и соответственно ***обновлять график тестирования***. Также производятся изменения графика тестирования, если планируемая функциональность не реализуется в данной версии. Имея в виду многообразие возможных изменений, требующих уточнения плана-графика, полезно создавать базовую версию плана-графика и версии при каждом последующем его принципиальном изменении. Нужно документально фиксировать каждое изменение плана-графика.

Особенности планирования тестирования сложных заказных комплексов программ

Жизненный цикл и ***планирование тестов комплекса программ*** должны проходить во времени, ***параллельно динамике изменения*** и жизненному циклу формирования требований и текстов программ. Тесты и сценарии их реализации должны быть адекватными и полностью отражать содержание компонентов комплексов программ, но в иной форме. Их следует рассматривать и анализировать, как еще одну форму описания функций КП, которое также может содержать дефекты и ошибки. Таким образом, программной (процедурной) форме представления комплекса программ должно полностью соответствовать его равноправное содержание в форме сценариев и тестов для проверки ***их взаимного соответствия***. При этом дефекты и ошибки возможны в обеих формах описания содержания, определение их места и устранение является основной задачей тестирования. При планировании тестирования целесообразно учитывать ***следующие рекомендации*** [12].

Тестирование в первую очередь проводить для требований с наивысшим приоритетом, которые представляют для заказчика наибольшую важность, либо которые могут причинить заказчику наибольшие неприятности в случае проявления дефектов программного продукта. Если запланировано тестирование всех требований, и ресурсы это позволяют, естественно, проверить выполнение всех требований. В случае недостаточных ресурсов, перед предъявлением продукта заказчику необходимо тщательно протестировать требования с наивысшими приоритетами.

Тестирование сначала осуществлять для новых функциональных возможностей, которые изменялись с целью исправления или совершенствования функций и характеристик. Если версия

является очередным обновлением или эксплуатируемой версией, особое внимание следует уделить новым функциям и характеристикам. Любые изменения, внесенные в программы, могут исказить даже те части комплекса программ, которые непосредственно не затрагиваются изменениями. В этом случае лучше всего как можно шире следует выполнять регрессионные тесты для всех функций и характеристик комплекса программ, какими бы ни были изменения.

Тестирование проводить тех компонентов и модулей, в которых наиболее вероятно присутствие дефектов. Если обнаружен какой-то дефект, часто рядом может быть еще несколько аналогичных дефектов. Если есть сведения от разработчиков, что часть компонентов породили проблемы во время модульного тестирования или проверки взаимодействия и функционирования компонентов, такие участки необходимо отметить, чтобы обратить на них внимание при комплексном тестировании или испытаниях.

Тестирование начинать с функций и конфигураций, с которыми наиболее часто будет иметь дело конечный пользователь или система. Для этого в спецификации требований полезно включать методику оценки случаев использования некоторых функций и иметь доступ к функциональному разрезу конечного пользователя – математическому ожиданию использования каждой функции. Это дополнительный источник информации для установки приоритетов тестирования. Большая часть времени, отведенного на тестирование, должна затрачиваться на проверку наиболее часто используемых функций, конфигураций и операций.

Планирование тестирования комплексов программ должно поддерживать:

- разработку множеств тестов на основе их проектирования сверху вниз, используя цели тестов, анализ рисков, требования к функциям и характеристикам комплекса программ;
- контроль корректности множества тестов за счет проектирования снизу вверх на основе реального функционирования и использования системы, и/или сценариев взаимодействия с пользователями;
- использование пробного, исследовательского тестирования для заполнения возможных пробелов в покрытии тестами требований функций и характеристик комплекса программ.

Планирование тестирования функций и характеристик комплекса программ должно содержать – рис. 2.10 [9, 12]:

- разработку автоматизированных и/или ручных тестов для покрытия всех функций, характеристик и рисков качества, которые определены как нуждающиеся в полном или сбалансированном ***приоритетном тестировании***, с особым вниманием к факторам функционирования системы, наблюдаемым со стороны пользовательского интерфейса или доступного программного интерфейса системы;
- тестовые данные или условия тестовых сценариев для покрытия ***критических вариантов*** использования системы, пользовательских данных или сокращения известных дефектов в программном продукте;
- тестирование в областях, которые ранее не были покрыты тестами, и которые, с точки зрения интуиции или результатов тестирования, представляются областями ***повышенного риска*** наличия дефектов;
- насколько позволяют время и ресурсы, использовать методы структурного покрытия для выявления не подвергавшихся тестированию областей и затем дополнять тесты для ***покрытия критических пробелов*** в тестовом покрытии;
- тестирование посредством ***методов изучения реализации*** программного продукта: тестирование потоков данных, тестирование путей и тестирование транзакций;
- ***опытную эксплуатацию*** и демонстрировать заказчику реальное применение – тестирование на основе того, что пользователи реально делают с комплексом программ и системой – бета-тестирование, изучение удобства использования и приемо-сдаточные испытания.

Управление требованиями к тестам включает в себя хранение требований, отслеживание связей, оценку рисков требований к тестам, выстраивание последовательности требований к тестам и определение методов верификации тестов. Отслеживание связей предполагает отображение тестовых процедур на требования к тестам и дефектов на тестовые процедуры. Заказчик и разработчики должны анализировать и, в конечном счете, ***утверждать план тестирования*** комплекса программ, в котором описаны требования к тестированию и представлена ***матрица соответствия системных требований и тестовых сценариев*** [3, 35]. Матрица соответствия должна содержать информацию о требованиях, а также показывать взаимосвязь между требованиями и другими результатами производства. Подпись заказчика под планом тестирования означает, что он утверж-

дает степень покрытия требований, проверяемых группой тестирования.

Требования к тестам должны заноситься в ***базу данных и/или матрицу отслеживания требований***. В базе данных или в матрице каждому требованию к тестам сопоставляется идентификационный номер компонента системной архитектуры программного продукта. Затем компонент архитектуры изучается вплоть до детализированных требований к программным компонентам и до системных требований или сценариев использования системы.

Матрица отслеживания требований тестами представляет собой технологический продукт, получаемый, возможно, автоматизировано при помощи инструмента управления требованиями. Она позволяет проследить системные требования и сценарии использования системы, а также покрытие требований тестовыми процедурами. Матрица может принимать одну из нескольких форм, основанных на различных видах отображений. ***Матрица отслеживания требований*** должна определять каждое требование, которое проверяется группой тестирования, а также метод его верификации. Поскольку матрица отслеживания требований определяет выполняемые тестовые процедуры, одобрение матрицы заказчиком также означает его удовлетворение степенью покрытия тестами системных требований или сценариев использования системы. При проведении приемосдаточных испытаний заказчик и тестировщики анализируют матрицу, чтобы проверить покрытие тестами системных требований или сценариев использования системы.

Глава 2.5

ТЕСТИРОВАНИЕ ПРИ ПРОИЗВОДСТВЕ ДИНАМИЧЕСКИХ ЗАКАЗНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Подготовка динамических тестов для тестирования заказных программных продуктов

Требования и реализация совокупности ***динамических тестов*** должны полностью ***обеспечивать возможность проверки выполнения утвержденных требований к программному продукту***, и соответственно должны быть адекватны по сложности и трудоемкости, производству такого программного комплекса. Эти ***два представления комплексов программ*** (требования и тесты) отличаются только формой описания их содержания: функциональным (процессным) или событийным (сценариями и результатами исполнения) (см. главу 2.2). Генерация тестов особенно сложна (также как и требований к программному продукту) для крупных заказных динамических комплексов программ ***реального времени***. Выбор типов моделей генерации тестов зависит от глубины знаний об алгоритмах функционирования комплексов программ, требований к характеристикам их качества и обобщенных параметрах необходимых результатов применения системы в целом. Кроме того, существенным для выбора типов моделей для генерации динамических тестов может быть длительность расчета имитированных тестовых данных и обеспечение возможности проводить полную обработку результатов тестирования.

В результате динамического тестирования сложных программных комплексов, необходимо достоверно устанавливать ***степень их соответствия утвержденным требованиям к функциям и характеристикам качества***. Для этого они должны проходить тщательные динамические испытания в условиях их применения. В реальных системах создание таких условий может быть очень дорого и опасно рисками при проявлении не устраненных при тестировании дефектов и ошибок, что недопустимо. Альтернативой является создание и применение ***математических моделей*** на компьютере, ***динамически***

имитирующих реальную внешнюю среду для генерации тестов и функционирования тестируемых комплексов программ. Таким образом, формируется возможность выполнять тестирование, выявлять и устранять ошибки, а также ***достигать высокого уровня соответствия программного продукта заданным требованиям***.

Одним из наиболее важных компонентов организации динамической верификации и тестирования является ***оценка трудоемкости и времени***, необходимых для их выполнения. Затраты на динамическое тестирование сложных КП могут составлять существенную часть (до 50%) стоимости крупного программного продукта, при этом важно для успеха этой операции, чтобы тестирование проводило достаточное число специалистов и у них было достаточно времени на качественное выполнение задач. Ограничения реальных ресурсов на верификацию и тестирование часто определяют ***достижимое качество динамических программных продуктов***.

Эффективная Программа динамического тестирования имеет свой собственный ***жизненный цикл разработки***, в который входят планирование стратегий и целей, определение требований к тестам, их анализ, проектирование и кодирование (рис. 2.12). Группа тестирования должна создавать специальные инструменты – ***генераторы динамических тестов***, которые автоматически подготавливают тесты. Автоматизация тестирования должна обеспечивать тестировщику возможность спроектировать и разработать на основе исходных требований к комплексу программ полный комплект тестовых сценариев, а затем с небольшими затратами ***повторять динамические тестовые сценарии*** для обнаружения и устранения ошибок

Затраты, необходимые для автоматизации генерации тестов, обычно больше затрат на ручное написание тестов. Для создания автоматизированных тестов, пригодных для процессов верификации и тестирования, должна быть выстроена инфраструктура, позволяющая тестировщикам определять действия, данные и ожидаемые результаты в удобном формате. Если решено вложить средства в автоматизацию тестирования, следует проанализировать последствия этого решения в рамках выбранной стратегии. При предположении исполнять автоматизированные тесты в регрессионных испытаниях или для целей технического обслуживания системы, может иметь смысл построить специализированный стационарный ***испытательный стенд*** генерации динамических тестов, который может находиться на рабочей

площадке испытателей весь период, пока поддерживается и развивается программный комплекс.

Подготовка средств тестирования динамических комплексов программ на соответствие требованиям включает:

- методы подготовки тестов для динамического тестирования сложных комплексов программ реального времени:
  - задачи автоматизированного формирования динамических тестов внешней среды;
  - оценки методов автоматизации разработки динамических тестов в реальном времени;
- компоненты генераторов динамических тестов внешней среды в реальном времени:
  - исходные данные для генерации тестов внешней среды;
  - состав средств для формирования тестов внешней среды;
- обработку результатов динамического тестирования комплексов программ в реальном времени:
  - средства оперативной обработки результатов динамического тестирования комплексов программ реального времени;
  - средства оценки характеристик комплекса программ и системы реального времени;
- оценки эффективности динамической генерации тестов в реальном времени:
  - оценки затрат на программную имитацию динамических тестов;
  - оценки экономической эффективности программной имитации тестов.

Рис. 2.12.

Метод структурного анализа, подразумевает детализированное описание архитектуры комплекса программ. ***Тестирование комплекса программ, основанное на архитектуре системы***, связывает тестовые процедуры с аппаратным обеспечением и компонентами архитектуры комплекса. Логика этой модели строится на понимании того, что аппаратное обеспечение и компоненты архитектуры системы могут быть отслежены вплоть до спецификаций системных требований и требований к программному продукту. Кроме того, формулировки требований к тестам можно отследить вплоть до компонентов архитектуры комплекса программ.

Программы динамического тестирования комплекса, в графической форме должны отражать его масштаб и методы, необходи-

мые для проведения тестирования. Определившись с моделью Программы тестирования, группа тестировщиков программного продукта должна разработать архитектуру тестирования. Структуру архитектуры тестирования можно представить двумя способами. Один метод организации тестовых процедур, известный как тестирование ***на базе архитектуры системы***, представляет тестовые процедуры компонентов системной архитектуры по логическому принципу. Второй метод – тестирование ***на базе методов*** увязывает тестовые процедуры с различными методами тестирования, представленными в модели Программы тестирования.

Важным аспектом организации динамического тестирования сложного программного продукта является решение о том, в каком объеме тестирование достаточно, и ***когда необходимо завершить процесс тестирования***. Тщательные измерения, такие как достигнутое покрытие тестами или охват функциональности, безусловно, очень полезны. Однако они не могут определить критериев достаточности тестирования. Принятие решения об окончании тестирования, включает анализ и учет стоимости и рисков, связанных с потенциальными сбоями и нарушениями надежности функционирования тестируемой динамической программной системы. В то же время, стоимость самого тестирования также является одним из ограничений, на основе которых принимается решение о продолжении тех или иных связанных с проектом работ (с частности, тестирования) или об их прекращении.

Для обеспечения высокого качества сложных заказных комплексов программ реального времени необходимы соответствующие ***проблемно-ориентированные интегрированные системы автоматизации динамического тестирования***, способные достаточно полно заменить испытания программ с реальными объектами внешней среды (см. рис. 2.12). При этом высокая стоимость и риск испытаний с реальными объектами почти всегда оправдывают значительные затраты на такие интегрированные системы, если предстоят испытания критических программ с высокими требованиями к качеству функционирования программного комплексов, с длительным жизненным циклом и множеством развивающихся версий. При необходимости такие генераторы могут быстро создавать тестовые данные, например, для проведения нагрузочного тестирования. Некоторые генераторы динамических тестовых процедур могут обладать высокой степенью интеграции в процесс анализа и проектирования программных

комплексов и позволять поэтапно тестировать их функции в соответствии со спецификациями требований и системной архитектурой. Другие генераторы могут извлекать информацию о документированных требованиях из архивов программных комплексов и создавать тестовые процедуры автоматически [3, 9, 35, 44].

Инструментальные средства автоматизации процессов динамического тестирования и испытаний сложных комплексов программ реального времени должны обеспечивать:

- определение и формирование динамических тестов – реализацию процесса тестирования разработчиком: ввод тестовых наборов; генерацию тестовых данных; ввод ожидаемых, эталонных результатов;
- выполнение участка тестируемого комплекса программ между контрольными точками, для которого средство тестирования может перехватить операторский ввод (клавиатуры, мыши и т.д.) и для которого вводимые данные могут быть отредактированы и включены в последующие тестовые сценарии;
- управление тестами и участком программы, для которого средство тестирования может автоматически выполнять тестовые наборы;
- анализ и обработку тестовых результатов – возможность средства тестирования автоматически анализировать части тестовые результаты: сравнение ожидаемых и реальных результатов; сравнение файлов; статистическую обработку результатов;
- анализ покрытия тестами исходных требований к программному продукту для обнаружения функций: которые не были выполнены; процедур, которые не были вызваны; данных, к которым не были обращения;
- анализ производительности комплекса программ, когда он исполняется: загрузку центрального процессора; загрузку памяти; обращения к специфицированным элементам данных и/или сегментам кода; временные характеристики функционирования испытываемой программы;
- моделирование внешней среды – поддержку процесса тестирования с помощью модели динамической имитации данных для программного комплекса из внешних аппаратных компонентов системы.

Методы ***динамического тестирования с исполнением контролируемого комплекса программ***, в большей или меньшей степени, ориентированы на обнаружение ошибок определенных типов,

преимущественно в структуре комплекса программ и реализуемых маршрутах обработки информации. Методы тестирования потоков данных ориентируются на выявление ошибок в вычислительной части программ и в процессах преобразования различной информации. Такая ориентация позволяет упорядочивать последовательность приоритетного применения методов с целью устранения, прежде всего, ошибок в наибольшей степени отражающихся на корректности исполнения программ, а также сосредоточиваться на методах, позволяющих решать частные задачи достижения необходимого их качества и соответствия требованиям при минимальных затратах.

Характеристики динамического функционирования программных комплексов зависят не только от их внутренних свойств, но и ***от свойств внешней среды***, в которой они применяются. Для сокращения неопределенностей и прямых ошибок при оценивании качества комплекса программ необходимо до начала испытаний определить основные параметры внешней среды и потоки информации, при которых он должен функционировать с требуемыми характеристиками при оценивании его качества и эксплуатации. Для этого заказчик и разработчик совместно должны ***структурировать, описать и согласовать модель внешней среды*** и ее параметры в среднем, типовом режиме применения, а также в наиболее вероятных и критических режимах, в которых должны обеспечиваться требуемые характеристики качества динамического функционирования программного комплекса. Такая ***модель должна отражать и фиксировать характеристики***:

- внешних, динамических потоков информации, в том числе, их распределение по видам источников, характеристикам качества данных и возможности их дефектов;
- интенсивность и структуру типовых сообщений от оперативных пользователей и администраторов, их необходимую квалификацию, отражающуюся на вероятности ошибок и качестве выдаваемой информации;
- возможных негативных и несанкционированных воздействий от внешней среды при применении программного комплекса;
- необходимые характеристики вычислительных средств, на которых предназначено функционировать комплексу программ с требуемым качеством.

При создании генераторов динамических тестов внешней среды применяется два принципиально различающихся подхода, которые

условно можно назвать интегральным и дифференциальным. При ***интегральном*** или эмпирико-статистическом подходе основой является формальное описание входной и выходной информации имитируемого динамического объекта, а также функциональной связи между данными на его входе и выходе. При этом структура объекта и процессы, реализующиеся при реальном функционировании его компонентов, не имеют значения и не моделируются. Исходные данные и характеристики для построения таких генераторов тестов получаются в натурных экспериментах или при исследовании более детальных – дифференциальных моделей.

В отличие от натурного эксперимента моделирование внешней среды и динамических тестов на компьютере имеет большие возможности контроля**,** как исходных данных, так и всех промежуточных и выходных результатов функционирования испытываемого программного комплекса. В реальных системах ряд компонентов иногда оказывается недоступным для контроля их состояния, так как либо невозможно поместить измерители контролируемых сигналов в реальные подсистемы, подлежащие тестированию, либо это сопряжено с изменением характеристик самого анализируемого объекта. Преимуществом моделирования внешней среды на компьютере является также ***повторяемость результатов ее функционирования*** и возможность исследования большого количества вариантов и сценариев тестирования. В отличие от этого натурные эксперименты зачастую невозможно остановить на некоторой промежуточной фазе или повторить с абсолютно теми же исходными данными.

Программная имитация динамических тестов внешней среды на компьютере в реальном времени позволяет [29, 33]:

- проводить длительное непрерывное генерирование имитируемых данных для определения характеристик функционирования комплекса программ в широком диапазоне изменения условий и параметров, что зачастую невозможно при использовании реальных объектов;
- расширять диапазоны характеристик имитируемых объектов за пределы реально существующих или доступных источников данных, а также генерировать динамические потоки информации, отражающие перспективные характеристики создаваемых информационных систем и объектов внешней среды;
- создавать тестовые данные, соответствующие критическим или опасным ситуациям функционирования объектов внешней среды,

которые невозможно или рискованно реализовать при натурных экспериментах;

- обеспечивать высокую повторяемость имитируемых данных при заданных условиях их генерации и возможность прекращения или приостановки имитации на любых фазах моделирования внешней среды.

Компоненты генераторов динамических тестов внешней среды в реальном времени

Одними из наиболее сложных и дорогих имитаторов внешней среды, применяемых для испытаний крупных программных комплексов, являются модели: полета космических аппаратов; диспетчерских пунктов управления воздушным движением; объектов систем противовоздушной обороны; сложных административных или банковских систем и другие. Применяемые для этого моделирующие испытательные стенды (МИС) проблемно – ориентированы и размеры программ, моделирующих в них динамическую внешнюю среду, могут даже ***значительно превышать размеры*** соответствующих испытываемых программных продуктов. Для их реализации должны выделяться достаточно мощные универсальные ***моделирующие компьютеры***. Кроме того, для автоматизации разработки программных комплексов могут использоваться отдельные ***технологические компьютеры***, что в совокупности образует инструментальную базу для обеспечения всего проектирования и производства сложных комплексов программ реального времени на ***объектных реализующих компьютерах***.

Имитация конкретных динамических тестов с реальными характеристиками, адекватными объектам внешней среды, является основной частью типовых ***моделирующих стендов*** (см. рис. 2.12). В соответствии с полной номенклатурой и реальными характеристиками таких объектов создаются их интегральные или дифференциальные модели. Выбор типов моделей зависит от глубины знаний об алгоритмах функционирования внешних объектов, характеристиках их компонентов и обобщенных параметрах работы объекта в целом. Кроме того, существенным для выбора типов моделей является длительность расчета имитированных данных и обеспечение возможности проводить полную имитацию динамической внешней среды на моделирующем компьютере ***в реальном времени*** с учетом затрат времени на обработку результатов.

Трудность адекватного моделирования некоторых динамических объектов внешней среды, особенно если при их функционировании активно участвует оператор-пользователь, не позволяет сосредоточить и полностью автоматизировать для крупных программных комплексов всю имитацию тестовых данных на моделирующих компьютерах. Поэтому при реализации интегрированных проблемно-ориентированных МИС для испытаний качества функционирования сложных программных продуктов приходится использовать аналоги реальных объектов внешней среды для формирования части данных. Разумное ***сочетание части реальных объектов внешней среды и имитаторов на компьютерах*** обеспечивает создание высокоэффективных МИС с комплексными моделями совокупностей объектов внешней среды, необходимых для динамических испытаний качества программных комплексов и систем в реальном времени. Такие стенды позволяют автоматическую генерацию тестов с помощью имитаторов на ЭВМ и аналогов реальной аппаратуры дополнять реальными данными от операторов пользователей, контролирующих и корректирующих функционирование систем обработки информации.

В схеме типового МИС можно выделить ряд базовых компонентов, назначение и функции которых представлены на рис. 2.13. Для каждого эксперимента при динамических испытаниях комплексов программ реального времени следует подготавливать план сценариев тестирования и ***обобщенные исходные данные.*** В моделирующем компьютере план и обобщенные исходные данные преобразуются в конкретные значения параметров для задания динамического функционирования каждого имитатора или реального объекта внешней среды.

Эти данные вводятся и преобразуются на моделирующем компьютере вне реального масштаба времени и подготавливают старт сеанса функционирования стенда и испытываемых программ в реальном времени. После этого начинают генерироваться тестовые данные.

Аналоги системы и аппаратуры внешней среды используются преимущественно для генерации динамических тестов, представляющих коррелированные логические переменные, которые трудно описать и смоделировать на компьютере. Кроме того, они позволяют проверить и аттестовать некоторые программные имитаторы внешней среды, которые впоследствии играют основную роль при испытаниях.

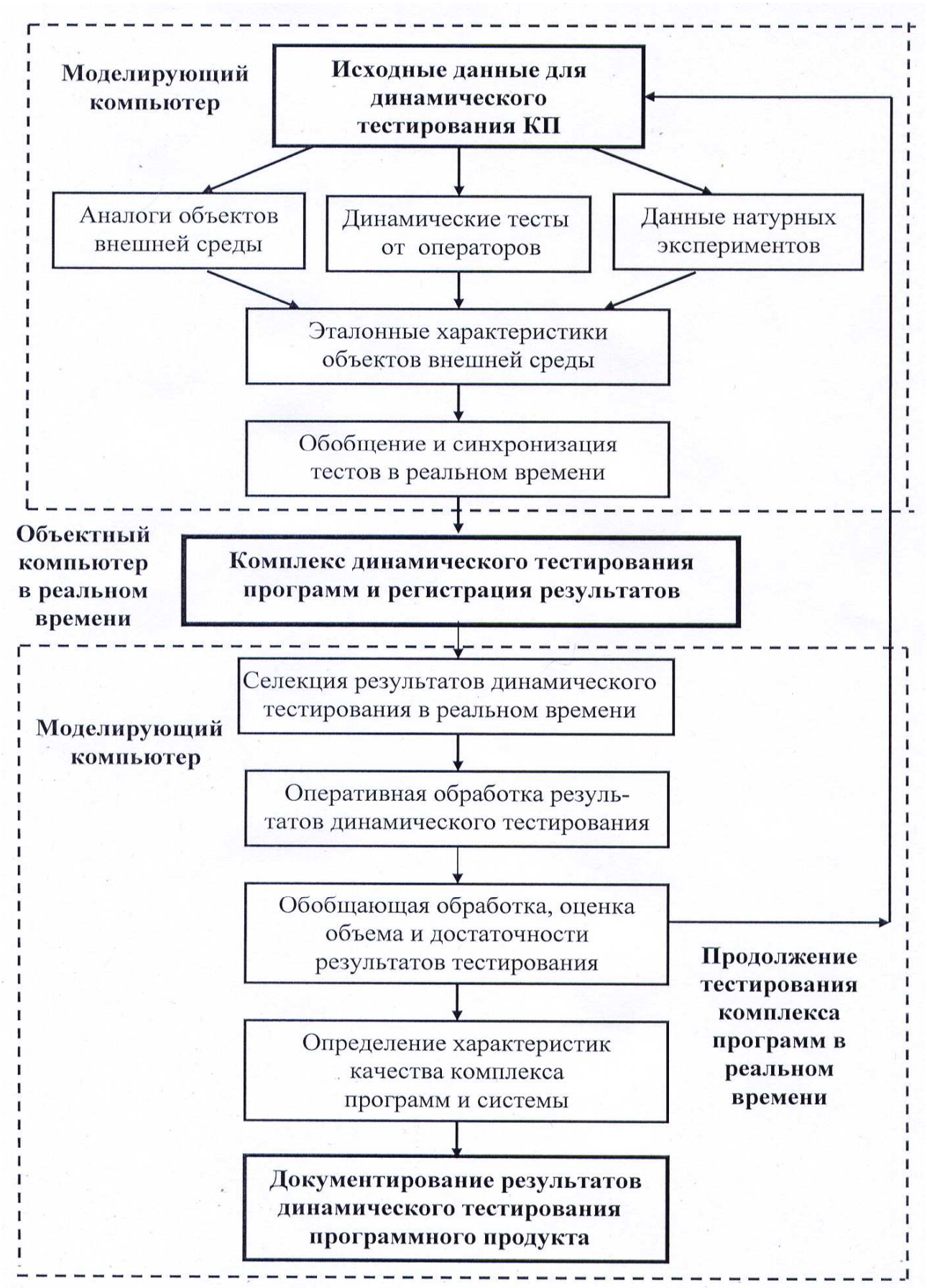
Моделирующий компьютер
Исходные данные для динамического тестирования КП
Аналоги объектов внешней среды
Динамические тесты от операторов
Данные натурных экспериментов
Эталонные характеристики объектов внешней среды
Обобщение и синхронизация тестов в реальном времени
Объектный компьютер в реальном времени
Комплекс динамического тестирования программ и регистрация результатов
Моделирующий компьютер
Селекция результатов динамического тестирования в реальном времени
Оперативная обработка результатов динамического тестирования
Обобщающая обработка, оценка объема и достаточности результатов тестирования
Определение характеристик качества комплекса программ и системы
Документирование результатов динамического тестирования программного продукта
Продолжение тестирования комплекса программ в реальном времени

Рис. 2.13.

В ряде случаев такие аналоги аппаратуры не могут отразить все особенности объектов внешней среды, и имитаторы на компьютере остаются единственными источниками соответствующей части данных для проверки качества программного комплекса.

Данные с рабочих мест операторов-пользователей должны отражать реальные характеристики динамических воздействий на тестируемый программный комплекс с учетом особенностей и квалификации человека, которому предстоит использовать испытываемые программы в реальной системе обработки информации. На эту часть МИС кроме первичных исходных данных от моделирующем компьютере могут вводиться данные обработки ряда тестов испытываемой системой. В результате через человека и его характеристики замыкается контур обратной связи ручного и автоматизированного управления динамическими объектами внешней среды. Такое же замыкание контура автоматизированного управления возможно в аналогах и аппаратных имитаторах реальных объектов.

Данные натурных экспериментов с объектами внешней среды могут подготавливаться заранее вне сеансов динамических испытаний программного комплекса, например, при отладке аппаратной части системы обработки информации. Эти данные отражают характеристики и динамику функционирования объектов, которые трудно или опасно подключать для непосредственного взаимодействия с недостаточно проверенными программами. Они могут быть полезны в тех случаях, когда создание определенных условий динамического функционирования объектов внешней среды очень дорого или опасно и может быть выполнено только в исключительных случаях. Однако данные натурных экспериментов не всегда удается адекватно описать условиями и обобщенными характеристиками их поведения.

При динамическом тестировании в ряде случаев необходимо иметь ***эталонные характеристики данных***, поступающих на испытываемый программный комплекс или систему. При работе с реальными объектами зачастую приходится создавать специальные измерительные комплексы, которые определяют, регистрируют и подготавливают к обработке все необходимые характеристики в процессе реального функционирования этих объектов (например, координаты движения самолетов при испытаниях систем УВД).

Синхронизация и обобщение тестовых данных предназначены для упорядочения динамических тестов от источников различных типов, в соответствии ***с реальным временем*** их поступления на ком-

плекс программ. В результате формируются потоки тестовых данных каждого реального объекта внешней среды, которые вводятся в объектный компьютер в соответствии с логикой функционирования системы обработки информации через соответствующие устройства сопряжения с моделирующим компьютером.

Повторяемость сеансов испытаний при автоматической имитации динамических тестов обеспечивается фиксированием всех исходных данных и применением программного формирования псевдослучайных чисел. При надежной работе аналогов реальных объектов и моделирующего компьютера, в принципе, можно добиться ***почти абсолютной повторяемости*** весьма длительных экспериментов и сценариев тестирования. Труднее обеспечивать повторяемость сценариев динамических испытаний, в которых активно участвует оператор-пользователь. В этом случае необходимо регистрировать и сохранять действия оператора в зависимости от времени, а затем повторять их в соответствии с записанным сценарием. При необходимости временная диаграмма может соблюдаться с точностью около 0,5-1 с, однако ошибки в действиях оператора и вводимых им параметрах могут отличаться в каждом сценарии тестирования.

Приведенные выше рекомендации по функциям и применению МИС ориентированы на создание ***сложных заказных программных комплексов***, их динамическое тестирование и испытания, в основном, до передачи в регулярную эксплуатацию. После приемки заказчиком или приобретения пользователями, в процессе функционирования и применения программного комплекса должно ***обеспечиваться их регулярное тестирование*** и оценка текущего качества. Для этого в ***составе программного продукта необходимы средства, обеспечивающие*** [9, 20, 35]***:***

- генерацию динамических тестовых сценариев или хранения тестов для контроля работоспособности, сохранности программного комплекса при функционировании и применении;
- оперативный контроль и обнаружение дефектов исполнения программ и обработки данных при использовании программного комплекса по прямому назначению;
- реализацию процедур предварительного анализа выявленных дефектов и оперативное восстановление вычислительного процесса, программ и данных (рестарт) после обнаружения аномалий динамического функционирования программного комплекса;

- мониторинг, накопление и хранение данных о выявленных дефектах, сбоях и отказах в процессе исполнения комплекса программ и обработки данных.

Средства генерации динамических тестов и имитации внешней среды в составе поставляемого заказного программного продукта предназначены для оперативной подготовки исходных данных при проверке различных режимов функционирования в процессе применения программного комплекса и при диагностике проявившихся дефектов. Минимальный состав средств генерации тестов должен передаваться пользователям для контроля использования рабочих версий в реальном времени и входить в комплект поставки каждой пользовательской версии программного комплекса. Более глубокие испытания функционирования версий и локализации ошибок следует проводить на базе средств ***имитации внешней среды высшего уровня в МИС на моделирующем компьютере***, которые используются специалистами для испытаний и/или сертификации системы.

Важной функцией испытательных стендов является их использование в качестве ***тренажеров для операторов-пользователей***. Так как качество функционирования комплексов программ может существенно зависеть от характеристик конкретного человека, участвующего в эксплуатации и обработке информации, то необходимо измерять эти характеристики. Необходимо также иметь возможность их улучшать до уровня, обеспечивающего выполнение ***заданных требований к программному продукту***. Поэтому в средства тестирования и испытаний органически должно входить обеспечение процессов динамической тренировки и измерения характеристик реального функционирования и реакции операторов, а также использование МИС для обучения и регулярной подготовки операторов-пользователей в процессе тиражирования и эксплуатации программного продукта.

Важнейшее значение для определения качества программных комплексов имеет ***адекватность имитаторов динамических тестов***, которая зависит от степени учета второстепенных факторов, характеризующих функционирование реальных объектов или источников информации, при создании их моделей. Точность моделей на компьютере, прежде всего, определяется алгоритмами, на которых они базируются, и полнотой учета в них всех особенностей моделируемых объектов. Кроме того, на адекватность имитации влияет уровень дефектов и ошибок в программах имитации. Каждый, не учиты-

ваемый в имитаторе элемент или фактор моделируемой системы, необходимо оценивать путем сопоставления частных имитируемых данных с результатами аналитических исследований или с данными, полученными на реальных системах, и определять его возможное влияние на полную требуемую точность модели и генерируемых динамических тестов с учетом других составляющих, отражающихся на достоверности имитации.

Обработка результатов динамического тестирования комплексов программ в реальном времени

Регистрация и обработка характеристик динамических тестовых данных должна обеспечивать их контроль на соответствие заданным требованиям к программному комплексу, обобщенным характеристикам каждого объекта внешней среды и исходным данным сеанса испытаний. Так проводится процесс испытаний по конечным результатам функционирования комплекса программ, выдаваемым внешним абонентам и для определения интегральных характеристик качества и их соответствия требованиям к программному комплексу. Однако для диагностики и локализации отказов, дефектов и ошибок, а также для оценки некоторых частных характеристик качества могут быть необходимы промежуточные данные исполнения программ на объектном компьютере.

Селекция результатов динамических испытаний может основываться на стратегии контроля функционирования программного комплекса ***снизу вверх***, т.е. от анализа исполнения отдельных компонентов программы и далее до стохастических результатов функционирования всего комплекса в динамике реального времени. При этом регистрируется избыточное количество данных, из которых затем отбирается минимум, необходимый для анализа. Может использоваться стратегия ***сверху вниз***, т.е. упорядоченное, иерархическое выделение в первую очередь обобщенных результатов функционирования комплекса программ с последующим уточнением регистрируемых и анализируемых результатов вплоть до детального контроля исполнения отмеченных программных компонентов. В этом случае регистрируются только те данные, которые необходимы для анализа в конкретном сеансе динамического тестирования. При обеих стратегиях необходимо иметь возможность управлять объемом и видом выделяемой и регистрируемой информации тестирования в зависимости от целей испытаний. Данные, получаемые и выделяемые в процессе испыта-

ний качества комплекса программ, целесообразно делить на следующие ***группы*** [20, 33, 35] :

- данные, характеризующие исходную тестовую информацию и выходные результаты динамического тестирования;
- маршруты исполнения программных компонентов и их операторов при некоторых фиксированных тестовых данных;
- аномальные события, сбои, отказы и данные, характеризующиеся отклонением результатов динамического тестирования от требований за допустимые пределы и ограничения;
- характеристики динамического использования различных ресурсов объектного компьютера.

Регистрация промежуточных данных обычно соответствует некоторым достаточно завершенным этапам испытаний функционирования программного комплекса. Вызовы регистрирующих программ должны подчиняться определенной системе контроля динамического функционирования программ при исходной гипотезе, что ***некоторые ошибки и дефекты в программах и данных могут проявиться на любой стадии тестирования.*** Однако количество вызовов регистрирующих программ и контроль промежуточных результатов, требующих нарушения целостности исполнения функциональных программ, следует ограничивать, учитывая допустимые расходы ресурсов времени на их реализацию. Так как основная задача регистрации при тестировании в реальном времени состоит в обнаружении и локализации ошибок и причин отказов с точностью до функциональной группы программ или компонента, то более точное определение места дефекта приходится переносить на тестирование компонентов и модулей по детерминированным сценариям вне реального времени.

Так как испытания современных крупных систем обработки информации и/или управления позволяют получать такое большое количество контрольных данных, что достаточно полный их анализ представляет трудную методическую и техническую задачу, ***обработку динамических результатов целесообразно осуществлять иерархически и дифференцировано*** (см. рис. 2.13). При избытке контролируемых величин снижается общее быстродействие имитаторов и комплекса программ в результате затрат времени на контроль и регистрацию. В каждом конкретном случае необходимо стремиться к компромиссу между полнотой регистрации промежуточных данных тестирования и удобством ***анализа обобщенных результатов на соответствие требованиям***.

Обработка результатов испытаний программных комплексов реального времени может быть разделена на две, достаточно автономные части: оперативную и обобщающую. ***Оперативная обработка результатов динамического тестирования*** должна производиться по упрощенным алгоритмам с большой пропускной способностью, обеспечивающим сохранение реального масштаба времени для всего испытываемого комплекса программ. Основная часть оперативной обработки результатов связана с замыканием контура обратной связи для имитации динамики функционирования управляемых объектов внешней среды. В оперативную обработку целесообразно включать расчет части интегральных данных динамического тестирования, позволяющих контролировать текущий процесс обработки информации испытываемым комплексом.

Обобщающая обработка накопленных результатов испытаний может производиться вне реального времени после завершения одного или серии испытаний. Основная задача при этом состоит в расчете различных интегральных характеристик качества функционирования программного продукта и ***соответствия их заданным требованиям заказчика***. Зарегистрированные и обработанные результаты испытаний должны использоваться для установления соответствия полученных характеристик качества ***заданным требованиям***. При выявлении их отклонения от требований технического задания заказчика, спецификаций требований или декларируемых в документации, должны разрабатываться корректировки программ для устранения несоответствия. Для этого все этапы тестирования и испытаний программных продуктов должны быть поддержаны системой конфигурационного управления версиями программных компонентов и базой данных документирования тестов, результатов испытаний и выполненных корректировок программ. Средства накопления сообщений об отказах, ошибках, предложениях на изменения, выполненных корректировках и оцененных характеристиках качества версий являются основой для конфигурационного управления развитием и совершенствованием комплекса программ.

Затраты на имитацию внешней среды и на генерацию динамических тестов для оценивания ***качества заказных программных продуктов*** могут быть одной из существенных составляющих при их создании. В ряде случаев, они соизмеримы с затратами на создание основных функций комплексов программ, что определяется принципиальным соответствием ***сложности необходимых наборов тестов***

и тестового покрытия программ, и ***сложности функций***, реализуемых испытываемым комплексом программ. Создание представительных совокупностей динамических тестов возможно путем использования реальных объектов внешней среды или с помощью программных имитаторов, адекватных этим объектам по результатам функционирования и генерируемой информации. При этом возникает проблема – ***какой метод и когда выгодней*** по затратам на генерацию тестов и по обеспечению необходимой степени покрытия тестами испытываемых комплексов программ.

Имитаторы тестов могут быть необходимы не только для оценивания достигнутых характеристик качества комплексов программ, но также для их комплексной отладки, квалификационного тестирования, испытаний, и при создания версий. Поэтому затраты на программные имитаторы и их экономическую эффективность целесообразно рассматривать в проекте с учетом всего комплекса задач, которые они способны и должны решать в жизненном цикле программного комплекса. Анализ эффективности программной имитации внешней среды при разработке и определении качества целесообразно разделять ***на две части***: оценка факторов, определяющих эффективность средств имитации тестов, и оценка экономического выигрыша при моделировании внешней среды на ЭВМ по сравнению с натурными экспериментами в реальных системах.

Факторы, определяющие эффективность программной имитации внешней среды на компьютере при разработке крупных комплексов программ, могут оцениваться в основном по их воздействию на качество создаваемых систем. Программная имитация внешней среды на компьютере может обеспечивать широкие наборы тестов и достаточно полные тестовые покрытия комплексов и компонентов при испытаниях, в том числе за пределами характеристик реально существующих или доступных источников тестов, а также соответствующие критическим или опасным ситуациям динамического функционирования объектов внешней среды [12, 17, 35].

Некоторые значения тестов не только трудно создать при натурных экспериментах, но они являются маловероятными в реальных условиях. Однако такие, даже маловероятные ситуации и значения тестов могут быть ***критическими и/или особо важными*** для функционирования всей системы, для которой разрабатывается и тестируется заказной программный комплекс. Выбор и имитация подобных ситуаций позволяют отрабатывать и оценивать качество в критиче-

ских маловероятных ситуациях, которые невозможно или опасно создавать на реальных объектах, но без их выполнения некоторые продукты не допустимо эксплуатировать в критических системах управления и обработки информации.

Экономическую эффективность программной имитации внешней среды на компьютере по сравнению с натурными экспериментами целесообразно оценивать при одинаковых объемах динамических тестовых данных для испытаний и определения качества. Показателем экономической эффективности имитации может служить ***соотношение затрат*** на проведение натурных экспериментов и затрат на программную имитацию той же совокупности тестовых и эталонных данных. Затраты ресурсов на натурные эксперименты для генерации тестов при проведении разработки, испытаний и определения качества пропорциональны реальному времени функционирования проверяемого программного комплекса и затратам на применение привлекаемых средств реальной внешней среды.

Затраты на программную имитацию динамических тестовых данных определяются ресурсами необходимыми на производство и эксплуатацию сложных комплексов программ для этих целей. Имитационные стенды практически всегда являются уникальными. В ряде случаев эти комплексы программ могут иметь объем порядка 10^6 строк текста и должны создаваться с применением современных технологических систем. Затраты на эксплуатацию программ имитации в основном определяются длительностью проведения динамического тестирования, испытаний и/или измерения характеристик качества.

Даже приближенные оценки при системном анализе соотношения этих затрат в большинстве случаев показывают ***высокую рентабельность программных имитаторов внешней среды***, особенно для квалификационного тестирования и оценивания характеристик качества крупных программных комплексов реального времени. Например, при тестировании системы для управления воздушным движением, применение имитационных стендов, по крайней мере, на порядок снижает затраты по сравнению с натурными экспериментами и использованием реальных объектов (самолетов), а для управления космическими аппаратами или атомными электростанциями это соотношение может быть значительно больше (~ 10 – 100).

Примером сложного испытательного стенда и моделей внешней среды для динамического тестирования и проверки ***на соответствие***

требованиям к функциям и характеристикам комплексов программ может быть ***тренажер управления полетами воздушных судов и диспетчерских систем в центрах управления воздушным движением***. Для комплексной отладки, тестирования, испытаний и сертификации программных продуктов управления воздушным движением (УВД) проводится имитация в реальном времени всей информации, поступающей из внешней среды. Источниками информации для центров УВД являются радиолокационные станции, летный состав на борту воздушных судов, диспетчеры управления воздушным движением и исходные планы полетов.

Тестирование надежности и безопасности функционирования программных продуктов в реальном времени

В составе требований к системам и программным комплексам, функционирующим в реальном времени, особое значение имеют ***динамические функции и характеристики.*** Для обеспечения их высокого качества непригодны отдельные сценарии и процедуры тестов, а необходимо создавать ***потоки динамических тестов в реальном времени***, адекватные соответствующим данным при функционировании внешней среды систем и/или пользователей. Эти потоки тестов должны обеспечивать динамическую проверку комплексов программ на соответствие требованиям, выявление дефектов и ошибок при реализации их функций и характеристик в реальном времени. Основная особенность такого тестирования состоит в необходимости создания динамической среды функционирования программного комплекса, максимально приближенной к реальной, при его практическом применении.

Задача состоит в определении соответствия требованиям, функций и характеристик программного комплекса при различной интенсивности потоков тестов, адекватных нормальным условиям применения программного комплекса, а также критическим по составу и интенсивности, для выявления предельных условий его работоспособности. Такие ***условия тестирования отражаются на интегральных характеристиках***, на снижении надежности и/или безопасности, а также на повышении рисков применения программного комплекса. Для комплексов программ реального времени ***особое значение могут иметь*** причины и методы уменьшения ***рисков надежности и производительности***, вследствие дефектов и ошибок, а также при формировании и реализации требований к этим характеристикам. Эти взаимосвязанные характеристики качества программных

комплексов зависят от одних и тех же свойств воздействий из внешней среды, требуют совместного анализа и методов для выявления и устранения дефектов. Локализация и устранение таких динамических дефектов обычно осуществляется вне реального времени, путем применения детерминированных сценариев и тестовых процедур, а иногда за счет изменения требований заказчика [17, 35,44].

Оценивание надежности программных комплексов включает измерение количественных характеристик: завершенности, устойчивости к дефектам, восстанавливаемости и доступности-готовности (см. главу 1.4). При этом предполагается, что в контракте, техническом задании или спецификации требований зафиксированы, и утверждены заказчиком определенные значения этих характеристик и их приоритеты. Измерения проводятся при функционировании готового программного комплекса для сопоставления с заданными требованиями и для оценивания рисков соответствия этим требованиям. Тестирование для оценки надежности комплекса программ должно проводиться в тестовом окружении, которое максимально приближено к реальным условиям применения системы. Входные параметры тестов следует задавать на основе вероятностного распределения соответствующих характеристик или их наборов при эксплуатации программного комплекса.

Для прямых, количественных измерений надежности необходимы инструментальные средства, встроенные в операционную систему или в соответствующие компоненты комплекса программ. Эти средства должны в динамике реального функционирования программ, автоматически селектировать и регистрировать аномальные ситуации, дефекты и искажения вычислительного процесса, программ и данных, выявляемые аппаратным, программно-алгоритмическим контролем или пользователями. Накопление и систематизация проявлений дефектов при исполнении программ позволяет оценивать основные показатели надежности, помогает определять причины сбоев и отказов и подготавливать данные для повышения надежности программных комплексов. Регулярная регистрация и обобщение таких данных способствует устранению ситуаций, негативно влияющих на функциональную пригодность и другие важные динамические характеристики.

Прямые экспериментальные методы оценивания интегральных характеристик надежности (безопасности и рисков), в ряде случаев весьма трудно реализовать при нормальных штатных условиях

функционирования крупных комплексов программ, из-за больших значений времени наработки на отказ (сотни и тысячи часов), которые необходимо достигать при разработке и фиксировать при испытаниях. Сложность выявления и регистрации редких отказов, а также высокая стоимость экспериментов при длительном многосуточном функционировании крупных комплексов программ приводят к тому, что на испытаниях получаются малые выборки зарегистрированных отказов и низка достоверность оценки надежности. Кроме того, при таких экспериментах трудно гарантировать полную представительность выборки исходных данных, так как проверки определяются конкретными условиями применения данного программного комплекса.

При испытаниях надежности в первую очередь обнаруживаются ***отказы – потери работоспособности.*** Однако в большинстве случаев первоначально остается неизвестной причина происшедшего отказа. Для выявления фактора, вызвавшего отказ (первичной ошибки или дефекта) и устранения его причины, необходимо, прежде всего, определить, каким компонентом системы стимулирован данный отказ. Для диагностики и устранения случайных редких отказов должна быть организована ***служба их регистрации*** с максимально полным фиксированием характеристик ситуаций, при которых проявился каждый. Для выявления ***тенденции изменения показателей надежности***, их зарегистрированные значения необходимо связывать во времени с моментами корректировки программ. Анализируя корреляцию между значениями надежности и процессом изменения программ, можно выявлять некоторые корректировки, которые содержат ошибки и снижают надежность.

При заключительных приемо-сдаточных и сертификационных испытаниях ***для достоверного определения надежности*** организуются многочасовые и многосуточные прогоны динамического функционирования комплекса программ в реальной и/или имитированной внешней среде в условиях широкого варьирования исходных данных с акцентом на стрессовые ситуации, стимулирующие проявления ***угроз надежности***. Такие прогоны позволяют измерять достигнутые характеристики надежности и определять степень их соответствия требованиям технического задания, а также закреплять их в технических условиях и документации на программный комплекс.

Если интенсивное тестирование программ в течение достаточно длительного времени не приводит к обнаружению дефектов или

ошибок, то у специалистов, ведущих испытания, создается ощущение бесполезности дальнейшего тестирования программного продукта, и он передается на эксплуатацию. Экспериментальное исследование характеристик сложных КП позволило оценить ***темп обнаружения дефектов – риски, при котором крупные программные продукты передаются на регулярную эксплуатацию***: 0,002–0,005 дефектов в день на человека, т.е. специалисты по испытаниям или все пользователи в совокупности выявляют только около одной ошибки или дефекта каждые два – три месяца использования КП. Интенсивность обнаружения ошибок ниже 0,001 ошибок в день на человека, т.е. меньше одной ошибки в год на трех-четырех специалистов, непосредственно выполняющих динамическое квалификационное тестирование и/или эксплуатацию комплекса программ, по-видимому, может служить ***эталоном высокой надежности*** обработки информации. Если динамическое функционирование программного продукта происходит непрерывно, то эти показатели соответствуют высокой наработке на обнаружение дефекта или отказа порядка 5 – 10 тысяч часов и коэффициенту готовности выше 0,99.

Форсированные (стрессовые) испытания для оценивания надежности (а также функциональной безопасности и рисков) программных комплексов значительно отличаются от традиционных методов испытаний аппаратуры. Основными факторами, влияющими на надежность, являются исходные данные и их взаимодействие с дефектами и ошибками программ или сбоями в аппаратуре компьютера. Поэтому форсирование испытаний надежности осуществляется повышением интенсивности искажений исходных данных и расширением варьирования их значений, а также специальным увеличением интенсивности потоков информации и загрузки программ на компьютер выше нормальной [9, 23, 33].

При форсированных испытаниях целесообразно выделять следующие ***режимы динамического тестирования***:

- полное искажение, предельные и критические значения ключевых параметров тестов каждого типа внешней информации и воздействий пользователей;
- предельные и критические сочетания значений различных взаимодействующих параметров тестов при эксплуатации программного комплекса;
- предельно большие и малые интенсивности суммарного потока и каждого типа динамической внешней информации;

• умышленные нарушения пользователями определенных положений инструкций и рекомендаций эксплуатационной документации на программный комплекс.

Особым видом форсированных испытаний является целенаправленное тестирование эффективности средств оперативного контроля и восстановления программ, данных и вычислительного процесса ***для оценивания восстанавливаемости***. При таких испытаниях основная задача состоит в оценивании качества динамического функционирования средств автоматического повышения надежности, и в измерении характеристик восстанавливаемости. Следует особо отметить трудности достижения и регистрации надежности программ, характеризующейся наработкой на отказ >>100 ч. При такой надежности резко возрастают сложность обнаружения возникающих отказов и диагностирования их причин.

В любых ситуациях функционирования сложных комплексов программ, прежде всего, должны исключаться катастрофические последствия и длительные отказы или в максимальной степени смягчаться их негативное влияние на результаты, выдаваемые пользователю. Предвидеть заранее все ситуации исполнения программ и протестировать при них крупные программные комплексы оказывается невозможным из-за их огромного количества, поэтому применяются методы, которые направлены на ***оперативное обнаружение последствий*** дефектов и аномального функционирования программ, а также на автоматическое восстановление (рестарт) нормального вычислительного процесса и искаженных текстов программ и данных. Для обеспечения высокой надежности функционирования необходимо максимально быстро обнаруживать аномалии, достаточно точно классифицировать тип уже имеющихся и возможных последствий искажений, а также осуществлять мероприятия, обеспечивающие быстрое восстановление нормального функционирования программного комплекса и системы.

Введение средств контроля функционирования и помехозащиты в программы позволяет ***скомпенсировать влияние на надежность неполной корректности программных комплексов***, а также снизить негативные воздействия внешних возмущений различных типов. Восстановление работоспособности при этом желательно производить настолько быстро, чтобы отказовую ситуацию можно было свести до уровня кратковременного сбоя.

Функциональная безопасность программных комплексов и систем зависит от отказовых ситуаций, негативно отражающихся на работоспособности и реализации их основных функций, причинами которых могут быть дефекты и аномалии в аппаратуре, комплексах программ, данных или вычислительных процессах. При этом катастрофически, критически или существенно искажается процесс функционирования программных комплексов и/или систем, что наносит значительный ущерб при их применении. Основными источниками отказовых ситуаций могут быть некорректные исходные требования, сбои и отказы в аппаратуре, дефекты или ошибки в программах и данных функциональных задач, проявляющиеся при их динамическом исполнении в соответствии с назначением.

Понятия, методы тестирование и характеристики функциональной безопасности программных продуктов и систем близки к аналогичным методам для надежности. Поэтому способы оценки и испытаний функциональной безопасности могут базироваться на методах тестирования, определения и обеспечения надежности функционирования комплексов программ. При более или менее одинаковых источниках угроз и их проявлениях эти понятия можно разделить ***по величине негативных последствий*** и ущерба при возникновении отказовых ситуаций. Чем сложнее системы и чем выше к ним требования безопасности, тем неопределеннее функции и характеристики тестирования требований для обеспечения их безопасности. Неопределенности начинаются с требований заказчиков, которые при формулировке технического задания и спецификаций ***не полностью формализуют*** и принципиально не могут обеспечить достоверное содержание всего адекватного набора характеристик и значений требований безопасности, которые должны быть при завершении проекта и предъявлении конечного программного продукта заказчику.

Всегда не полностью, с необходимой детализацией определены и описаны все характеристики, особенности функционирования и ***безопасности объектов внешней среды***. Эти характеристики в той или иной степени обычно находятся под воздействием управляемой системы. Квалификация и субъективные свойства потребителей и пользователей изменяются по мере освоения функциональных возможностей системы и ее работоспособности, что увеличивает неопределенность ее реальной безопасности. Различия свойств персонала, применяющего систему, дополнительно увеличивают неопределен-

ность значений безопасности и трудности ее прогнозирования при тестировании с учетом множества ***субъективных факторов различных специалистов, участвующих в эксплуатации***.

При анализе характеристик функциональной безопасности целесообразно выделять и учитывать особенности ***двух классов заказных систем и их программных продуктов.*** Первый класс составляют системы, имеющие встроенные комплексы программ жесткого регламента реального времени, автоматизировано управляющие внешними объектами или процессами. Время необходимой реакции на отказовые ситуации таких систем обычно исчисляется секундами или долями секунды, и процессы восстановления работоспособности должны проходить за это время, в достаточной степени автоматизировано (бортовые системы в авиации, на транспорте, в некоторых средствах вооружения, системы управления атомными электростанциями). Системы второго класса, применяются для управления процессами и обработки деловой информации из внешней среды, в которых активно участвуют специалисты-операторы (банковские, административные, штабные военные системы). Допустимое время реакции на опасные отказы в этих системах может составлять десятки секунд и минуты, и операции по восстановлению работоспособности частично могут быть доверены специалистам-администраторам по обеспечению функциональной безопасности.

Эти факторы влияют на ***неопределенность критериев, методов и оценивания эффективности тестирования функциональной безопасности*** конкретных программных комплексов и систем. Существующие технологии тестирования способствуют повышению функциональной безопасности, снижению потенциального ущерба и рисков, однако практически всегда остается открытым вопрос, насколько применяемые методы оправдывают затраты на реализацию требований заказчика.

Достижение требуемой функциональной безопасности систем, содержащих программные продукты реального времени, решается путем использования ***современных регламентированных технологических процессов динамического тестирования***, подобных применяемым при обеспечении надежности. Они должны быть поддержаны группой международных стандартов, определяющих состав и процессы выполнение требований к заданной функциональной безопасности систем и комплексов программ.

Тестирование производительности и использования ресурсов компьютеров программными продуктами

Оценивание ресурсной эффективности состоит в измерении количественных характеристик: временной эффективности и используемости динамических ресурсов ЭВМ комплексом программ. При этом предполагается, что в контракте, техническом задании и спецификации требований зафиксированы и утверждены требуемые значения этих характеристик и их приоритетов. При тестировании производительности должны применяться промышленные нагрузки, что позволяет предсказать поведение программного комплекса и системы при реальной эксплуатации. Средства, обеспечивающее тестирование производительности, должны позволять оценивать влияние перегрузок.

Оценивание перегрузок – это процесс тестирования работоспособности вычислительных машин при обработке большого потока данных с целью выяснения того, когда и где программный комплекс выйдет из строя под высокой нагрузкой. Эти ***допустимые пределы должны быть определены в системных требованиях*** к программному комплексу, где также должна определяться реакция системы на перегрузки. Данный вид тестирования необходим для систем, работающих с максимальной спроектированной нагрузкой, для проверки того, что они динамически функционируют в соответствии с требованиями.

Адекватное проведение динамического ***тестирования программного комплекса на перегрузки***, при использовании ручных методов подготовки тестов – дорого, трудоемко, неточно и занимает много времени. Выделение значительных ресурсов для тестирования требует больших затрат, и трудно организовать совместную работу необходимого числа пользователей и компьютеров. Необходимы средства автоматизированного тестирования, которые включают имитаторы нагрузки и позволяют тестировщикам динамически имитировать виртуальных пользователей или объектов, одновременно работающих с целевым программным продуктом.

Для ***измерения характеристик временной эффективности*** необходимы инструментальные средства, встроенные в операционную систему или в соответствующий комплекс программ. Эти средства должны в динамике реального функционирования программного комплекса регистрировать:

- загрузку вычислительной системы функционирующими программами;
- значения интенсивности потоков данных для обработки от конкретных внешних абонентов;
- длительность исполнения типовых заданий при реализации конкретных функций;
- характеристики функционирования устройств ввода/вывода;
- время ожидания результатов (отклика) на функциональные задания пользователей или системы;
- заполнение памяти обмена с внешними абонентами в различных режимах применения программного комплекса.

Значения этих характеристик зависят не только от свойств и функций комплекса программ, но также от особенностей архитектуры и операционной системы компьютера. Регулярная регистрация и обобщение таких данных позволяет выявлять ситуации, ***негативно влияющие*** на функциональную пригодность, надежность и другие важные характеристики программного комплекса. Существует особый вид тестов для проверки удовлетворения специфических требований, предъявляемых к ***параметрам производительности***. При этом может производиться динамическое тестирование с целью достижения реальных (достижимых) ***возможностей*** по производительности, и функционирования программ с повышением нагрузки.

При высокой интенсивности поступления исходных данных может ***нарушаться временной баланс*** между длительностью решения требуемой совокупности задач программным комплексом в реальном масштабе времени, и производительностью компьютера при решении этих задач. Также возможно нарушение баланса между имеющейся в компьютере памятью и памятью, необходимой для хранения всей поступившей и обрабатываемой информации. Для выявления подобных ситуаций и определения характеристик программного комплекса в условиях недостаточности ресурсов компьютера проводятся испытания при высокой, но допустимой, в соответствие с требованиями, интенсивности поступления исходных данных.

При использовании комплексом программ производительности и памяти реализующем компьютером менее чем на 50%, разработчик может практически не учитывать эти ограничения и сопутствующие риски. Закономерным является стремление разработчиков программ применять, особенно для систем реального времени, встроенные объектные компьютеры ***с предельным использованием их технических***

характеристик. Опыт создания программ реального времени позволяет утверждать, что практически невозможно использовать производительность объектного компьютера более чем на 95%, и почти всегда целесообразно ограничиваться на уровне 80 – 90%, так как иначе, затраты на разработку программного комплекса могут значительно увеличиться. Для оценивания характеристик использования производительности при тестировании крупных программных продуктов ***должны быть измерены***:

- реальные значения интенсивностей поступающих исходных данных и заданий на вызов функциональных программ, а также распределения вероятностей этих интенсивностей для различных источников и типов заданий;
- длительности автономного решения отдельно каждой функциональной задачи, обрабатывающей исходные данные или включаемой внешними заданиями, а также периодически;
- загрузка компьютера в нормальном режиме поступления сообщений и заданий, а также вероятность перегрузки заданиями различных типов и возможные распределения длительностей перегрузки в реальных условиях;
- влияние пропуска в обработке заданий или сообщений каждого типа и снижения темпа решения определенных задач на функциональную пригодность и другие важные характеристики программного комплекса.

Перечисленные задачи могут быть ***решены экспериментально*** в процессе тестирования завершенного разработкой программного комплекса, однако при этом ***велик риск***, что производительность компьютера окажется недостаточной для решения заданной совокупности задач в реальном времени, что отразится на качестве использования системы. Поэтому при оценивании требуется принимать специальные меры для создания реальных, а также контролируемых, наиболее тяжелых по загрузке условий функционирования комплекса программ и внешней среды. Такие ***критические ситуации*** могут быть в значительной степени предотвращены в процессе разработки комплекса программ путем расчета длительностей исполнения компонентов по тексту программ, и объединения этих характеристик в соответствии со структурой всего комплекса программ.

Для корректного ***оценивания предельной пропускной способности*** системы с данным программным комплексом необходимо из-

мерять следующие характеристики реализации функциональных программ в процессе разработки:

- экстремальные значения длительностей их исполнения и маршруты, на которых эти значения достигаются;
- среднее значение длительности исполнения каждой функциональной группы программ на всем возможном множестве маршрутов и его дисперсию;
- распределение вероятностей и значений длительности исполнения функциональных групп программ.

Для оценивания ресурсной эффективности, при подготовке технического задания и спецификаций требований ***следует согласовывать с заказчиком модель и характеристики внешней среды***, в которой будет применяться комплекс программ, а также динамику приема и передачи данных. Для определения использования комплексами программ временных ресурсов компьютеров полезно применять рекомендации стандарта **ISO 14756**. Стандарт ориентирован на динамическое оценивание: программных комплексов, операционных систем и вычислительных комплексов, включающих аппаратные и программные средства.

По результатам квалификационного тестирования и испытаний могут быть решены задачи динамического оценивания ресурсной эффективности программного комплекса, что позволяет анализировать факторы, определяющие необходимую пропускную способность компьютера, и разрабатывать меры для приведения ее в соответствие с потребностями. Если предварительно в процессе проектирования производительность системы не оценивалась или определялась слишком грубо, то, ***велик риск***, что доработки будут большими или может понадобиться заменить компьютер на более быстродействующий. Это обусловлено, как правило, «оптимизмом» разработчиков, что приводит к ***занижению интуитивных оценок длительностей решения функциональных задач*** и возможных предельных интенсивностей потоков внешней информации.

Глава 2.6

СОПРОВОЖДЕНИЕ СЛОЖНЫХ ЗАКАЗНЫХ ПРОГРАММНЫХ КОМПЛЕКСОВ

Организация и методы сопровождения сложных программных комплексов

В процессе эксплуатации версий программного продукта у каждого пользователя могут появляться некоторые претензии к функционированию, которые квалифицируются им как ошибки или дефекты эталонной (базовой) или собственной версии. От пользователей или заказчика могут поступать также предложения по внесению изменений в базовую версию для улучшения эксплуатационных характеристик и расширения функциональных возможностей системы и комплекса программ. Аналогичные предложения могут поступать от разработчиков. Для общения с пользователями и накопления информации о выявляемых недостатках в тиражируемых сложных КП, целесообразно выделение группы специалистов высокой квалификации, овладевших всеми функциями системы и программного продукта.

Целью сопровождения является выявление и устранение обнаруженных дефектов и ошибок в программах и данных, введение новых функций и компонентов в КП, анализ состояния и корректировка документации, тиражирование и контроль распространения версий, актуализация и обеспечение сохранности документации и физических носителей. Основная задача – изменить и улучшить существующий программный продукт, сохраняя его целостность и функциональную пригодность. Для сохранения и повышения качества сложных комплексов программ ***необходимо регламентировать процессы модификации и совершенствования программных средств***, а также поддерживать их соответствующим тестированием и контролем качества.

Широкое применение прототипирования и повторного использования готовых апробированных программных компонентов способствовало превращению сопровождения в особый раздел методов и средств обеспечения жизненного цикла КП. Технология сопровож-

дения должна обеспечивать координированное развитие множества версий и их компонентов, каждая из которых имеет достаточно высокое качество и специфические функции, а также, возможно, различных пользователей. Благодаря этому со временем программные средства должны улучшаться и совершенствоваться, как по функциональным возможностям, так и по качеству решения каждой задачи.

При организации сопровождения сложных КП следует учитывать важные ***психологические факторы, усложняющие привлечение и деятельность менеджеров и квалифицированных специалистов*** в этой области [1, 21, 31]:

- эта деятельность требует очень высокой квалификации и больших умственных затрат, связанных, прежде всего, с необходимостью одновременного, широкого охвата и анализа множества компонентов и их взаимосвязей, находящихся в различных состояниях завершенности модификаций;
- корректируемые компоненты зачастую разрабатывались в прошлом в разное время, различными специалистами, в различном стиле и с неодинаковой полнотой документирования, что усложняет освоение их содержания при внесении изменений и устранении дефектов;
- сложная, творческая сторона работ при сопровождении вуалируется тем, что приходится овладевать и анализировать программы, разработанные ранее другими специалистами, которые зачастую может быть проще не корректировать, а разработать заново;
- комплексы программ, прошедшие широкие испытания и эксплуатацию у заказчиков гарантируют достигнутое качество результатов функционирования, и любые в них изменения имеют высокий риск внесения дополнительных ошибок и ухудшения этого качества, что ограничивает возможность коренных модификаций;
- выполняемые работы требуют особой, скоординированной тщательности корректировок и четкого регламентированного взаимодействия ряда специалистов, различающихся квалификацией и уровнем ответственности;
- процессы и результаты сопровождения не отличаются наглядностью и внешним эффектом, проявлением их размера и сложности, вследствие чего не престижны среди рядовых программистов и недооцениваются руководителями проектов.

По мере развития применения сложных заказных программных продуктов стало ясно, что интегральные затраты на их сопровождение и создание новых версий ***могут значительно превосходить затраты на разработку*** их первой версии. Опыт последних лет показал, что во многих случаях для сопровождения и мониторинга версий необходимо практически такое же, или даже большее, число специалистов, чем разработало первую версию КП. При создании сложных КП перемещение специалистов с разработки новых программных компонентов на развитие и сопровождение версий имеет систематический характер. Это приводит к тому, что по мере накопления эксплуатируемых КП и их компонентов все большее число специалистов переходит из области непосредственного программирования новых программ в область системного проектирования и создания новых версий ***на базе повторно используемых компонентов***.

Только после завершения создания нескольких версий может прекратиться переход дополнительных кадров в сферу сопровождения и управления конфигурацией и установиться стабильное соотношение между числом специалистов, занятых первичной разработкой новых проектов и сопровождением версий. Очень часто разработчики новогоКП не предусматривают этот процесс и требующиеся ресурсы, что значительно снижает эффективность последующего применения созданного программного продукта.

Для определения и оценки качества модифицированного программного средства могут быть использованы различные показатели, качественные и количественные стандартизированные метрики в соответствии со стандартом **ISO 9126**. Сопровождаемость должна быть определена до начала первичной производства программного продукта, соответствующим ***соглашением между заказчиком и разработчиком-поставщиком***. Разработчик должен подготовить план обеспечения сопровождения, в котором отражены конкретные методы, соответствующие ресурсы и последовательности работ. Требования к процессам сопровождения определяются группой основных факторов, влияющих на реализацию модификации программных средств, образующих концептуальную цепочку: ***требования на изменения – изменяемые функции – размер (масштаб) изменений – стратегия модификаций – ресурсы необходимые для их реализации.*** Эта логическая схема обычно используется при последовательном анализе процессов сопровождения сложных КП. При этом ос-

новным критерием оценки сопровождения является ***совершенствование функциональной пригодности*** и улучшение характеристик качества программного продукта.

Процесс сопровождения в соответствии со стандартом **ISO 14764**, использует основные процессы производства и тестирования комплексов программ, а также вспомогательные процессы документирования, управления конфигурацией, обеспечения качества, верификации, аттестации, совместного анализа, аудита и устранения дефектов. Стоимость процесса сопровождения может составлять значительную (даже наибольшую) часть стоимости жизненного цикла программного продукта. Период значительного изменения размера, функций и характеристик качества в крупных проектах заказных комплексов программ составляет обычно 1-2 года. В результате исследований появилось понятие «критической сложности и расширения размера» модифицируемой части версии при сопровождении. Если при модернизации и выпуске очередной версии размер доработок заметно превышает «критический», то велика вероятность частичного ухудшения характеристик системы или необходимости выпуска нескольких промежуточных версий для устранения ошибок в изменениях и достижения высокого качества проведенной модернизации.

Заказчик может заключить соглашение с разработчиком базовой версии КП об ***организации сопровождения*** или выбрать в качестве сопроводителя третью сторону (помимо разработчика). Сопровождение может также быть проведено по соглашению между двумя сторонами внутри одного предприятия. Эти положения должны быть использованы независимо от того, принадлежит ли заказчик или поставщик к одному или к разным предприятиям. ***Передача программного средства*** для сопровождения является контролируемой и координируемой последовательностью действий, при выполнении которых разработанный продукт переходит от предприятия, выполнившего его первоначальное производство, к специалистам или предприятию, проводящему его сопровождение. В процессе передачи должны быть отражены: требования к передаче технических и программных средств, данных и знаний (опыта) от разработчика к сопроводителю; задачи, необходимые для реализации стратегии сопровождения программного продукта: комплектование персонала, его обучение, ввод в действие версий программного продукта, распространение опыта по сопровождению.

Сопроводители иногда встречаются с необходимостью сопровождать программный продукт ***с минимальным набором документов*** или даже при отсутствии их. При отсутствии необходимых документов сопроводитель должен их создать, что является обязательной частью полного корректного сопровождения. В подобной ситуации, сопроводитель при подготовке к сопровождению должен:

- определить проблемную область (тип программного продукта);
- изучить любые доступные документы, по возможности обсудить программный продукт с разработчиками и поработать с данным продуктом;
- изучить структуру и организацию программного продукта;
- провести его инвентаризацию, подвергнуть продукт управлению конфигурацией, выстроить продукт в соответствии с библиотеками управления конфигурацией, создать сценарии тестирования и проанализировать структуру данного продукта;
- определить функции, реализуемые программным продуктом; по возможности рассмотреть технические требования (спецификации) к данному продукту, его общую структуру, предоставить данный продукт другим сопроводителям и прокомментировать программные коды;
- установить приоритеты предложений о модификации комплекса программ.

Сопроводитель и заказчик должны ***заключить договор на сопровождение*** и указать в нем возможные процедуры внесения изменений в сопровождаемые программные продукты. Процедуры могут быть использованы как разработчиком оригинала, базовой версии КП, так и независимым сопроводителем и ***охватывать***:

- основные требования и правила, используемые для определения того, когда КП может быть локально откорректирован, а когда необходима новая базовая версия программного продукта с использованием для ее подготовки и инсталляции процессов производства;
- описания типов редакций версий, в зависимости от частоты их появления или влияния на эксплуатацию программного продукта (например, экстренные редакции, периодические редакции);
- способы информирования заказчика о состояниях текущих или намечаемых изменений;
- классификацию типов изменения, его очередности (приоритетности) и взаимосвязи с другими предложенными изменениями.

Персонал сопровождения должен проводить проверку внесенного изменения совместно с заказчиком, утвердившим модификацию в целях подтверждения функциональной пригодности и работоспособности откорректированного программного продукта и получить подтверждение того, что внесенное изменение удовлетворяет требованиям сопровождения, установленным в договоре.

Разработку концепции сопровождения, целесообразно начинать с формализации и обоснования набора исходных данных, отражающих общие особенности класса, назначения и функции КП, потребителей и этапов жизненного цикла проекта, каждый из которых влияет на выбор определенных характеристик изменения комплекса программ. Их описания желательно предварительно упорядочить по приоритетам с учетом особенностей назначения, сферы модификации и применения конкретного программного продукта. На этапе создания концепции и системного анализа следует сформировать цели сопровождения, выбрать методы и алгоритмы модификации основных, функциональных задач, а также сформулировать предварительные критерии качества создаваемых новых программных компонентов и данных. При этом, естественно, встает вопрос о ресурсах, которые потребуются для достижения этих целей, и о возможности их реализации.

В ***концепции сопровождения*** заказчик и специалисты-разработчики должны представить требования, документально оформить планы и процедуры для проведения работ и реализации задач этого процесса. Они должны определить процедуры для получения, документирования и контроля сообщений о дефектах и заявок на внесение изменений от пользователей, а также для обеспечения обратной связи с пользователями. Всякий раз, когда возникают проблемы (дефекты), они должны быть документально оформлены и введены в процесс решения. Для анализа и ликвидации их следует реализовать процесс управления изменениями и конфигурацией существующего КП и определить организационные процедуры взаимодействия с данным процессом. Необходимо проанализировать сообщение о дефектах и заявках на внесение изменений по их влиянию на организационные процессы, существующую систему и интерфейсные связи с другими системами и установить:

- корректировку, модернизацию, профилактику или адаптацию к новым условиям;

- размер изменения, стоимость, время на реализацию изменения;
- критичность, влияние на основные функции, производительность, безопасность или защиту.

На основе проведенного анализа персонал сопровождения должен разработать варианты реализации процессов изменения и документально оформить: сообщение о дефекте или заявку на модификацию; результаты их анализа и требования к реализации изменений. Следует согласовать с заказчиком выбранные варианты изменений в соответствии с договором.

Описание концепции сопровождения должно быть первым шагом при разработке политики сопровождения КП. Она определяется наличием соответствующих ***ресурсных и бюджетных ограничений и должна охватывать*** [9, 19, 35]:

- типы допустимых изменений и процедур сопровождения;
- уровень качества сопровождаемых документов;
- реакцию (чувствительность) пользователей на сопровождение;
- обеспечиваемый уровень обучения персонала сопровождения;
- обеспечение поставки модифицированных версий программного продукта;
- возможность организации справочной службы – «***горячей линии***».

Концепция должна учитывать задачи сопровождения программного продукта после поставки очередной версии КП. Важной частью концепции сопровождения является ***определение ресурсов и специалистов*** (физических или юридических), отвечающих за сопровождение продукта. Должна быть проведена ***оценка условий финансирования и стоимости сопровождения***. Стоимость зависит от размера области сопровождения. Дополнительными факторами, подлежащими учету, являются стоимость: обучения, как сопроводителей, так и пользователей; среды программного инструментария, тестирования и их ежегодного сопровождения. Разработку и утверждение в концепции ***спецификаций требований к функциональным характеристикам и качеству программного продукта*** с учетом выполненных изменений, целесообразно проводить итерационно. Полная и однократная формализация требований к характеристикам каждой крупной модификации в начале жизненного цикла КП обычно невозможна, прежде всего, из-за разных представлений заказчика и разработчиков

о деталях её назначения, функций и возможностей реализации при доступных ресурсах

В зависимости от сложности программного продукта окончательным результатом работ при прогнозировании изменений комплекса программ должны быть детализированные и утвержденные требования к номенклатуре, свойствам и значениям качества программного продукта, которые достаточны для его полноценного сопровождения и последующей, эффективной эксплуатации. Эти ***требования закрепляются в концепции, контракте и техническом задании***, по которым сопроводитель впоследствии должен отчитываться перед заказчиком при завершении модификаций. Однако на последующих этапах жизненного цикла и при конфигурационном управлении, требования могут изменяться по согласованию между заказчиком и разработчиком, которые чаще всего приурочиваются к подготовке новой базовой версии. Для этого необходим мониторинг функциональной пригодности, масштаба проекта, требований и реализаций характеристик в течение всего ЖЦ .

Принципиальные и технические возможности, точность реализации свойств и измерения значений характеристик КП, а также общие ресурсы конкретного проекта всегда ограничены в соответствии с их содержанием и возможностями заказчика и разработчиков. Это определяет ***рациональные диапазоны значений каждого изменения***, которые могут быть выбраны в концепции сопровождения программного продукта на основе требований заказчика, здравого смысла, а также путем анализа пилотных проектов и прецедентов в спецификациях требований реализованных модификаций.

Этапы и процедуры при сопровождении сложных заказных программных комплексов

В соответствии с требованиями стандартов по развитию и модификации программного продукта в жизненном цикле ***должен быть организован процесс и этапы его сопровождения***, который включает:

- подготовку процесса сопровождения КП;
- анализ проблем и изменений;
- внесение изменений;
- проверку и приемку изменений при сопровождении версии КП;

- перенос компонентов на иную аппаратную и операционную платформу;
- снятие с эксплуатации КП.

Эти разделы и соответствующие процессы, детализированы в стандарте **ISO 14764**. После активизации процесса следует разработать план сопровождения и соответствующие процедуры, а также выделить конкретные ресурсы для сопровождения. После поставки заказчику программного продукта сопроводитель, в соответствии с договором и предложением о модификации или отчетом о дефектах, должен изменить соответствующие программы и документы. Исходные данные преобразуют или используют в работах по сопровождению для получения выходных результатов – ***модифицированных версий программного продукта***. Рекомендуется проводить регулярный контроль с целью проверки корректности выходных результатов конкретных работ по сопровождению.

При ***подготовке процесса изменений*** сопроводитель должен создать план и определить процедуры, выполняемые при реализации сопровождения. План сопровождения целесообразно создавать параллельно с планом разработки первой, базовой версии программного продукта. При выполнении данной работы сопроводитель также должен определить необходимые организационные интерфейсы и взаимосвязи между специалистами и с другими предприятиями. Исходными данными для работ по подготовке процесса являются: старая (исходная) базовая версия программного продукта; системные документы; предложения о модификациях и отчеты о дефектах. Для обеспечения эффективной реализации процесса сопровождения сопроводителю следует разработать и документально оформить ***стратегию проведения сопровождения***.

Сопроводитель должен разработать, документально оформить и выполнять планы и процедуры для проведения работ и решения задач процесса сопровождения. В плане сопровождения следует описать стратегию сопровождения системы, а в процедурах сопровождения должны быть определены подробности выполнения этапов и процессов сопровождения. Для обеспечения создания эффективных планов и процедур сопровождения сопроводитель должен:

- выполнить оценку сопровождаемой системы;
- гарантировать официальное подтверждение принятия на себя обязанностей сопроводителя программного продукта;
- провести анализ доступных ресурсов для сопровождения;

- оценить и согласовать с заказчиком финансирование и стоимость сопровождения;
- установить требования к процессу передачи программного продукта сопроводителю;
- определить подлежащие реализации процессы сопровождения;
- документально оформить процесс сопровождения в виде планов и процедур, согласованных с заказчиком.

Целью планирования сопровождения является подготовка плана работ по сопровождению и обеспечение ресурсов, необходимых для проведения этих работ после передачи программного продукта на сопровождение. Планирование начинается после определения концепции сопровождения и завершается разработкой плана сопровождения, используемого в качестве основы при сопровождении. ***Общий план сопровождения должен определять***:

- причины необходимости сопровождения;
- состав исполнителей работ по сопровождению;
- роли и обязанности каждого субъекта, вовлеченного в сопровождение;
- как должны быть выполнены основные процессы и работы;
- какие имеются и необходимы ресурсы для сопровождения;
- методы и средства организации работ по управлению, выпуску продукта и синхронизации работ;
- перечень всех проектных результатов и продуктов, подлежащих поставке заказчику;
- критерии завершения соответствующей деятельности, работ и задач;
- состав отчетных материалов по этапам, затратам и графикам проведения работ;
- периодичность и способы выдачи отчетных материалов;
- состав отчетных материалов по проблемам и устраненным дефектам;
- время начала и длительность сопровождения.

Проектирование архитектуры модификаций определяет функции и компоненты модифицированного программного средства. Основными особенностями данной работы среди процессов ЖЦ, влияющими на сопровождаемость, являются выбор структуры программы, разбиение её на компоненты (модули) и поток данных, циркулирующий между ними. При модификациях важно использовать

знания коллектива специалистов по обработке данных, относящиеся к возможности использования частей существующих программ или библиотек, доказавших высокое функциональное качество. Основными средствами, способствующими обеспечению требований сопровождаемости, являются модульная архитектура в сочетании с нисходящим анализом и соответствующие документы, в которые при необходимости могут быть внесены дополнения.

Каждая возникающая проблема и дефект должна быть документально оформлена и введена в процесс анализа изменений, для чего следует:

- разработать схему классификации и присвоения приоритетов для предложенных модификаций и описаний дефектов;
- разработать процедуры проведения целевых анализов изменений;
- определить процедуры представления предложенных модификаций и описаний дефектов оператором;
- определить организацию обратной связи с пользователями при анализе изменений;
- определить, как пользователей будут обслуживать в период реализации сопровождения КП;
- определить, как будут введены предлагаемые модификации в базу данных учета состояний изменений и используемых ресурсов.

До внесения изменений в систему и в КП сопроводитель должен: проанализировать возможные изменения с точки зрения их влияния на деятельность предприятия, существующую систему и взаимосвязанные с ней системы; разработать и документально оформить рекомендуемые альтернативные решения по внесению корректировок и согласовать принятое решение по внесению изменений с заказчиком. Сопроводителю необходимо проанализировать отчет о проблеме – дефекте или предложение о модификации по их влиянию на организационные вопросы, существующую систему и интерфейсные связи с другими системами.

Для обеспечения реализации представленного предложения на изменение сопроводитель должен определить [9, 19, 33]:

- наличие соответствующих специалистов, способных реализовать предлагаемое изменение;
- наличие достаточного финансирования для реализации предлагаемого изменения в программе;

- наличие соответствующих ресурсов компьютера, и степень влияния модификации на реализуемую версию, программного продукта;
- влияет ли отсутствие предполагаемых изменений на требования к системным интерфейсам, ожидаемый срок службы системы, приоритеты;
- влияние изменений на безопасность и защиту системы при эксплуатации;
- преимущества, получаемые после проведения модификации;
- влияние реализации изменений на графики проведения работ по версии программного продукта;
- необходимые процессы верификации, тестирования и оценки характеристик системы и программного продукта после внесения корректировки.

Для того чтобы подтвердить актуальность представленных отчетов о дефектах, сопроводитель должен ***продублировать и верифицировать*** возникшие проблемы – дефекты, выполнив следующие этапы решения данной задачи: разработать стратегию верификации и квалификационного тестирования для проверки устранения конкретной проблемы – дефекта; провести тестирование для проверки наличия проблемы – дефекта; документально оформить результаты квалификационного тестирования.

Сопроводитель должен реализовать ***процесс управления конфигурацией*** для управления изменениями существующей системы или определить организационный интерфейс с данным процессом. Результатами данной работы являются: план и процедуры сопровождения; процедуры решения проблем и устранения дефектов; планы организации обратной связи с пользователями; план передачи модификаций заказчику и пользователям. До внесения изменений в систему и программный продукт в соответствии с договором с заказчиком, сопроводитель должен ***согласовать выбранный вариант корректировки***.

В конце работ должен быть проведен ***анализ рисков***. На основании выходных результатов анализа может быть пересмотрена предварительная оценка требуемых ресурсов и с привлечением пользователей или заказчика принято решение о целесообразности перехода к работе по внесению изменений в базовую версию программного продукта. Результатами этой работы являются: анализ влияния измене-

ний на риски; рекомендуемый вариант и согласованные изменения; обновленные и исправленные документы.

Результаты испытаний корректировок должны быть документально оформлены. Контроль за рассматриваемой работой должен быть проведен посредством процесса совместного анализа. Результатами данной работы являются: обновленные планы, документы и процедуры тестирования; измененные исходные программы; отчеты о квалификационном тестировании; показатели, характеризующие качество внесенных изменений. Обновленные документы должны включать: подробный отчет о проведенном анализе; обновленные требования; обновленные планы, процедуры и отчеты о тестировании; обновленные учебные материалы.

Проверка и приемка модификаций при эксплуатации обеспечивает подтверждение корректности изменений, внесенных в систему, в соответствии с принятыми стандартами и по установленной методологии. Проверки проводятся для гарантирования правильности изменений и их согласованности с точки зрения выполнения установленных требований заказчика к программному продукту. Сопроводитель должен провести проверки каждого внесенного изменения совместно с заказчиком, утвердившим изменение в целях подтверждения целостности и работоспособности измененной системы:

- отслеживание реализованных предложений о модификации и отчетов о дефектах относительно требований предыдущей базовой версии проекта и программных кодов;
- проверку тестируемости текста (кодов) программы;
- проверку соблюдения стандартов на ЖЦ КП и системы;
- проверку того, что изменены только нужные компоненты программного средства;
- проверку правильности сборки новых компонентов программного продукта;
- контроль обновления документов версии программного продукта;
- проверку полноты проведения тестирования и отчетов о тестировании.

Результатами данной работы являются: новая базовая версия заказного программного продукта, включающая в себя принятые изменения; отклоненные изменения; отчет о приемке версии; отчеты о проверках и аудитах; отчет о квалификационном тестировании программного продукта. Большую роль для успешного внедрения новых

версий играет ***психологический аспект***. Благоприятные условия внедрения обеспечиваются там, где имеется нормальное ***взаимодействие заказчика, пользователей и разработчиков*** во время создания и изменения версий заказного программного продукта. Это способствует достаточно высокой степени отработки документации, инструментальных средств разработчиками и своевременному уяснению функционального назначения компонентов КП, его особенностей и новых возможностей пользователями.

Сопроводитель должен ***документально оформить*** и представить заказчику:

- отчеты о проблемах (дефектах) и предложения о модификациях; результаты их анализа и варианты реализации изменений;
- результаты приемочных испытаний, верификации, аттестации и измерений характеристик качества новой версии программного продукта;
- отчеты об обеспечении характеристик качества программного продукта и результаты их тестирования;
- результаты аудиторских проверок версии программного продукта;
- замечания заказчика и результаты взаимодействия с ним по устранению дефектов версии программного продукта;
- комплект актуальных проектных документов и документов результатов сопровождения;
- оценки корректности реализованной политики, графика и Программы квалификационного тестирования версии программного продукта;
- соотношение оценок необходимых и использованных ресурсов;
- официальные рекомендации с указаниями о целесообразных последующих модификациях и создании новых версий КП.

Внедрение новой версий программного продукта для массового применения осуществляется, как правило, в два этапа; силами разработчиков модификаций в целях обкатки, проверки и выявления ошибок в изменениях на стадии опытной эксплуатации, и посредством использования специализированных коллективов сопровождения для тиражирования и распространения. Сопроводители в этом случае получают возможность непосредственно контролировать работу пользователей с системой и документацией, что обеспечивает высокую оперативность отработки замечаний и рекламаций, формирова-

ние квалифицированных предложений для изменений, оценку эффективности применения версии программного продукта.

Снятие программного средства с эксплуатации и сопровождения должно быть подготовлено анализом, обосновывающим это решение. В анализе следует определить и экономически обосновать: возможность сохранения устаревшей версии комплекса программ, а также необходимость создания и применения новой версии программного продукта. Специалисты, выполняющие снятие программного продукта с сопровождения и эксплуатации, должны разработать план, предупредить пользователей об этом, провести соответствующее обучение персонала, уведомить всех заинтересованных субъектов о завершении сопровождения и архивировать соответствующие данные. В ***содержание документов*** необходимо включить:

- анализ требований заказчика к снятию с сопровождения и эксплуатации;
- оценку влияние снятия с сопровождения программного продукта на систему;
- установить программный продукт, заменяющий снимаемый (при его наличии);
- график и Программу снятия программного продукта с сопровождения и эксплуатации;
- определить и документировать все процедуры по снятию с сопровождения и эксплуатации;
- сроки прекращения полной или частичной поддержки сопровождения;
- требования по архивации версии и модификаций программного продукта и соответствующих документов;
- сроки перехода, при необходимости, к новой версии программного продукта;
- требования по доступу к архивным копиям данных проекта программного продукта.

Ресурсы для обеспечения сопровождения сложных заказных программных комплексов

Прогнозирование необходимых основных ресурсов – труда, времени и числа привлекаемых специалистов для сопровождения сложных комплексов программ осложнено тем, что затраты на изменения состоят из двух, принципиально различных частей. Первая, обычно наименьшая, часть изменений, характеризуется затратами на

обнаружение и устранение дефектов и ошибок в КП, проявление которых ***не предсказуемо*** и имеют большие флюктуации в зависимости от характеристик проекта, квалификации специалистов, применяемого инструментария, и ряда других трудно учитываемых факторов. Априори оценить и прогнозировать такие затраты при сопровождении конкретных КП вряд ли возможно и ниже они не рассматриваются. Вторая часть изменений регламентирована целеустремленным совершенствованием и упорядоченными модификациями версий программного продукта, масштаб которых ***может предварительно прогнозироваться*** с некоторой достоверностью. Такие изменения могут служить основой для определения возможных затрат на разработку дополнительных функций и значительных модификаций версий КП, которые можно обобщать на некотором интервале времени сопровождения или для проекта в целом.

При анализе затрат на сопровождение программных продуктов целесообразно рассматривать следующие ***сценарии***:

- определение размера отдельных локальных модификаций программ и данных практически без учета взаимодействий с остальной частью версии программного продукта, благодаря его четкой структурированности и возможности выделения размера конкретной изменяемой функции;
- совокупные затраты ресурсов на реализацию каждой модификации, имеющей глубокие взаимосвязи с множеством компонентов всего сложного программного продукта, при которой необходимо учитывать не только размер конкретного изменения, но и степень влияния на весь комплекс программ;
- оценивание интегральных затрат и совокупных размеров изменений при сопровождении и управлении конфигурацией КП в течение некоторого интервала времени (месяц, год) с учетом всего множества изменений.

Первым этапом прогнозирования необходимых ресурсов при сопровождении является создание ***комплекса требований*** к конкретной модификации функций программного продукта, на которые могут быть разбиты фактические компоненты, определяющие функциональную пригодность КП. В дальнейшем разбиение может детализироваться, формируя упрощенный или более точный уровень абстракции и взаимодействия изменяемых компонентов. Следует учитывать, что в максимальной степени детализированная структура КП может принести пользу на стадии предварительного оценивания размера

модификации. Один из путей оценки размера изменений КП, находящегося на этапе концепции проекта, заключается в сравнении его функциональных задач и свойств с уже существующими версиями. При обосновании необходимых ресурсов для сопровождения сложных КП наибольшее значение имеют ***три ключевых фактора***:

- размер – масштаб, подлежащих разработке полностью новых или модификаций программных компонентов;
- размер и относительная доля готовых программных компонентов, которые могут быть заимствованы из предшествовавших проектов, и повторно использованы, в очередной версии программного продукта;
- относительные затраты ресурсов на создание модификаций и новых компонентов с оцененным масштабом изменений: труда специалистов, времени, бюджета на единицу размера (на строку изменения текста программ) или полные затраты на разработку всей новой версии программного продукта.

Эти факторы могут быть оценены квалифицированными экспертами на основе имеющегося у них опыта мониторинга и реализации предшествовавших подобных модификаций. Достоверность прогнозов требующихся ресурсов зависит, прежде всего, от ***точности оценки исходных требований на совершенствование программного продукта***. Они позволяют использовать опыт прошлых разработок и их отличия от новых методов и функций, предусмотренных в конкретных проектах, а также индивидуальные возможности коллектива разработчиков или другие уникальные особенности конкретно проекта. При наличии перечисленных исходных данных и положительной оценке целесообразности экспертного анализа и прогнозирования ресурсов для сопровождения КП их следует использовать для:

- оценки размера – масштаба (числа строк) предполагаемого изменения текста разрабатываемых новых программ, с учетом размера готовых повторно используемых компонентов;
- расчета возможной полной трудоемкости и длительности разработки корректировок версий, а также среднего числа специалистов, необходимых для их реализации;
- обобщения основных технико-экономических показателей и ***оценки полной стоимости сопровождения КП***, анализа результатов, и обоснования, рентабельности продолжения мониторинга, модификаций и сопровождения комплекса программ.

Приступая к разработке модификаций комплекса программ, необходимо сначала провести реалистическую оценку возможного изменения ***масштаба проекта*** – поставленных целей, ресурсов проекта и выделенного времени. Задача управления масштабом состоит в задании базовых требований, которые включают разбитое на компоненты ограниченное множество дополнительных функций и требований, намеченных для реализации в конкретной версии проекта. ***Базовый уровень изменений масштаба КП должен обеспечивать***:

- приемлемый для заказчика минимум дополнительных функций и требований к версии программного продукта;
- разумную вероятность успеха с точки зрения возможностей коллектива сопровождения и разработчиков модификаций за требуемое время.

Для ***уменьшения возможных методических ошибок*** оценок ресурсов для сопровождения и модификации КП следует начинать с прогнозирования размеров изменений или новой версии программного продукта, достоверность и ошибки которых могут быть обусловлены ***следующими факторами***:

- проблема, цель и новые дополнительные функции КП могут быть недостаточно хорошо поняты разработчиками модификаций и/или заказчиками из-за того, что некоторые существенные факторы были упущены или искажены из-за предвзятого к ним отношения;
- специалисты-оценщики масштаба изменений версии КП могут допустить значительные ошибки при попытке описания того, насколько большими могут быть изменения системы или комплекса программ, до этапа разработки концепции или предварительного проекта модификации;
- менеджеры и специалисты проекта полагают, что желательно прогнозировать возможные изменения масштаба и спецификаций требований в начале сопровождения, заказчики же часто считают, что не стоит тратить драгоценное время на оценки размеров предстоящих модификаций и на детальную разработку требований к сопровождению.

Уникальный ***заказной программный продукт***, основная часть жизненного цикла которого приходится на производство, может создаваться почти без учета последующих затрат на сопровождение. Однако многократно модернизируемые, и широко тиражируемые продукты требуют больших затрат на сопровождение. Вследствие длительного срока сопровождения и эксплуатации (в ряде случаев более

10 лет), а также большого числа версий, содержащих результаты модернизаций, совокупные затраты на сопровождение в некоторых случаях значительно превышают затраты на первичное производство комплекса программ. Эти затраты распределяются по всему интервалу времени сопровождения, вследствие чего при подготовке каждой версии затраты обычно меньше, чем на первичную разработку КП. Длительное сопровождение иногда вызывает неоднократную смену специалистов, осуществляющих сопровождение. При таких заменах появляются значительные ***затраты на обучение*** новой группы сопровождения, что вызывает рост общих затрат.

Затраты на обнаружение и устранение дефектов и ошибок в программе при сопровождении определяются двумя факторами: затратами на обнаружение каждой ошибки и затратами на устранение выявленных ошибок при формировании очередной версии. При сопровождении непрерывно требуются затраты для контроля состояния версий программ и обеспечения их сохранности. Широко тиражируемый комплекс программ объемом ~ 10^6 строк, может требовать непрерывных усилий коллектива в составе десятка и более специалистов для устранения ошибок, корректировок версий и документации.

Затраты на совершенствование и модернизацию программ близки по содержанию (но не по величине) к затратам на их первичную разработку. Модернизация обычно производится поэтапно. Для каждой новой эталонной версии изменяется (разрабатывается) только некоторая часть от всего объема КП. Эта часть при вводе очередной версии может составлять 10 – 20% от объема всего комплекса. Сложность связей в КП приводит к тому, что удельные затраты на изменяемые программы, при модернизации каждой версии могут быть в 2 – 3 раза больше, чем затраты на создание программ такого же объема при первичном проектировании. Эта величина зависит от того, насколько путем стандартизации архитектуры и интерфейсов, при системном проектировании предусматривались перспективы совершенствования и модификации КП. Для выполнения этих работ иногда используется коллектив специалистов, осуществивших первичную разработку программного продукта. Такая организация наиболее характерна для ***уникальных заказных*** продуктов.

Затраты на тиражирование каждой новой версии включают совокупные затраты на производство очередных экземпляров программного продукта, их инсталляцию в объектных компьютерах и освоение для нормальной эксплуатации. Затраты на тиражирование

версий при сопровождении практически пропорциональны произведению числа версий и их тиража. Вследствие этого даже относительно малые затраты на каждый экземпляр при внедрении новой версии могут приобретать большое значение в жизненном цикле всей серии версий заказного программного продукта.

На основе анализа и оценивания рассчитанных характеристик ресурсов для сопровождения следует выполнять заключительное ***технико-экономическое обоснование необходимости сопровождения конкретного программного продукта*** и определять:

- целесообразно ли продолжать работы по сопровождению и мониторингу конкретного программного продукта или следует его прекратить, вследствие недостаточных ресурсов специалистов, времени или большой трудоемкости разработки модификаций;
- при наличии достаточных ресурсов, следует ли провести маркетинговые исследования для определения рентабельности создания очередной версии программного продукта и поставки её заказчику;
- достаточно ли полно и корректно формализованы концепция и требования к модификациям версий программного продукта, на основе которых проводились экспертные оценки и расчеты затрат, или их следует откорректировать и выполнить повторный анализ с уточненными исходными данными;
- есть ли возможность применить готовые повторно используемые компоненты КП, в каком объеме относительно размера комплекса программ и рентабельно ли их применять в конкретной версии программного продукта или весь проект целесообразно разрабатывать как полностью новый.

Глава 2.7

УПРАВЛЕНИЕ КОНФИГУРАЦИЕЙ И ДОКУМЕНТИРОВАНИЕ СЛОЖНЫХ ЗАКАЗНЫХ ПРОГРАММНЫХ КОМПЛЕКСОВ

Процессы управления конфигурацией программных комплексов

Цель управления конфигурацией при производстве и сопровождении сложных программных комплексов и систем, состоящих из многих компонентов (единиц конфигурации – ЕК, каждая из которых может иметь разновидности или версии), обеспечить управляемое и контролируемое развитие их структуры, состава компонентов и функций, а также сокращение дефектов в течение всего жизненного цикла КП. В процессе организации конфигурационного управления необходимо построить и использовать компактные и наглядные ***схемы однозначной иерархической идентификации и изменения взаимодействия компонентов КП. Процессы управления конфигурацией***, применяют административные и технические процедур на всем протяжении ЖЦ программных комплексов для: обозначения, определения и установления состояния версий программных продуктов в системе; управления изменениями и выпуском компонентов и комплекса; описания и сообщения о их состояниях и заявок на внесение изменений. Эти процессы включают: подготовку процессов; определение конфигурации; учет состояний конфигурации; оценку конфигурации; контроль конфигурации; управление выпуском и поставкой программного продукта. Все основные и вспомогательные процессы подлежат адаптации и конкретизации применительно к характеристикам определенного проекта [34, 41].

Изменения конфигурации КП и его компонентов в ***плане управления проектом*** должны предусматривать ***действия*** с четкими разделами:

- ***почему*** и с какой целью производится корректировка программ или данных;
- ***кто*** санкционирует и выполняет изменения комплекса программ или компонентов;
- ***какие*** действия и процедуры должны быть выполнены для

реализации изменений каждой единицы конфигурации;

- ***когда*** по срокам и в координации с какими другими процедурами следует реализовать определенную модификацию компонентов и конфигурацию программного продукта;
- ***как*** и с использованием каких средств и ресурсов должны быть выполнены запланированные изменения компонентов и программного продукта .

Четкая ***организация и автоматизация этого процесса*** позволяют избегать множества вторичных ошибок, обусловленных недостаточной координацией проводимых корректировок, и формирование новых версий сложных КП коллективом специалистов. Этому должна способствовать утвержденная иерархия принятия решений на изменения компонентов и комплекса в целом должностными лицами проекта (см. таблицу 2.2), поддержанная методами и средствами защиты от несанкционированного доступа при выполнении корректировок специалистами различной квалификации и права доступа на разных уровнях проекта.

Стандарт **ISO 15846** обобщил, детализировал и развил основные ***концептуальные положения управления конфигурацией***. Существенным достоинством этого стандарта является подробное и систематичное изложение практических рекомендаций по управлению конфигурацией сложных комплексов программ, которые целесообразно использовать в современных заказных проектах систем. В процессе проектирования КП должна быть формализована и документально зафиксирована, ***концепция организации конфигурационного управления программным комплексом***, содержащая:

- ожидаемую длительность поддержки производства и модификации конкретного КП;
- масштаб и уровень предполагаемых изменений и модификаций КП;
- возможное число и периодичность выпуска версий программного продукта;
- организационные основы процессов сопровождения и конфигурационного управления программным комплексом;
- требования к документированию изменений версий КП;
- кто будет осуществлять управление конфигурацией – заказчик, покупатель, разработчик или специальный персонал поддержки ЖЦ комплекса программ.

Если предполагается, что программный продукт будет иметь длительный жизненный цикл или ожидаются значительные изменения КП, то следует рассмотреть и учесть наиболее детальные требования к методике организации и к коллективу, ответственному за конфигурационное управление и его документирование. В стратегии управления следует учесть характеристики системы: количество компонентов программного комплекса, типы, размер и критичность создаваемых и применяемых программных компонентов и продуктов. Управление конфигурацией (УК) следует организовать так, чтобы персонал знал свои обязанности и имел достаточно независимости и полномочий для выполнения поставленных задач.

Управление конфигурацией включает действия и средства, позволяющие устанавливать категории, статус и личности руководителей, которые правомочны определять целесообразность и эффективность изменений, а также техническую реализуемость корректируемых версий с учетом ограничений бюджетов и сроков (рис. 2.14). При анализе и селекции изменений важен точный учет степени влияния каждого изменения на все остальные компоненты и на их основные характеристики качества. Поэтому решения о кардинальных изменениях комплекса и компонентов должны приниматься руководства проектом на достаточно высоком уровне, способного оценить их влияние на концептуальную целостность и качество всей информационной системы (см. таблицу 2.2).

Концепция конфигурационного управления конкретным проектом должна предусматривать ***анализ изменений иерархической структуры – конфигурации*** программных комплексов и их компонентов, как сверху вниз, так и снизу вверх. Первая задача состоит в обеспечении пошаговой детализации анализа сверху вниз возможных причин конкретных дефектов (проявлений ***вторичных ошибок***) или неэффективности функционирования программы, для обнаружения их первичного источника (***первичной ошибки***) (см. главу 2.2). Вторая задача при движении анализа снизу вверх должна обеспечивать проверку корректности взаимодействия связанных корректировок и сохранения концептуальной целостности и качества комплекса программ и его компонентов. Средства и методы УК должны быть ориентированы на ***координированное развитие множества версий КП и их компонентов***, каждая из которых имеет достаточно высокое качество и специфические функции.

Управление конфигурацией компонентов и комплексов программ включает:

- методы управления конфигурациями требований и тестов компонентов и комплексов программ:
  - конфигурационную идентификацию и учет версий компонентов и комплексов программ;
  - управление запросами на изменения версий компонентов и комплексов программ;
  - реализацию корректировок версий компонентов и комплексов программ;
- процессы и средства управления конфигурациями требований и тестов компонентов и комплексов программ:
  - контроль корректности изменений конфигурации версий компонентов и комплексов программ;
  - сборку и формирование версий конфигурации программного продукта;
  - методику оформления отчетов о выявленных дефектах, ошибках и предложениях по корректировке требований и тестов;
  - создание отчетов о состоянии и изменениях конфигурации версий программного продукта;
  - утверждение и выпуск версий конфигурации программного продукта заказчику и пользователям;
  - архивирование и тиражирование версий программного продукта и документов.

Рис. 2.14.

Корректировки могут проводиться не синхронно по множеству версий заказчика, в результате чего образуется набор версий выполненных изменений, учитывающих текущие состояния версий продукта у конкретных пользователей.

Важной целью управления конфигурацией является документальное оформление и обеспечение полной наглядности текущей конфигурации программ и данных и степени выполнения требований к их функциональным характеристикам. Другая задача заключается в том, чтобы все лица, работающие над КП, в любой момент его жизненного цикла использовали достоверную и точную информацию о всех единицах конфигурации (ЕК) и их взаимодействии.

Таблица 2.2.

Типы изменений требований к компонентам и продукту, право на их выполнение и утверждение

Объекты изменения требований	**Типы изменений требований объектов**	**Право на выполнение изменения требований имеют:**	**Право на утверждение изменения требований имеют:**
Требования к версиям модулей и компонентов программ и документов	Устранение дефектов в требованиях к программным модулям и компонентам	Программисты – разработчики требований к модулям и компонентам	Руководители разработки требований к функциональным группам программ
Требования к версиям функциональных компонентов программ и документов	Корректировки требований к функциям и взаимодействию программных компонентов	Руководители разработки требований к функциональным компонентам	Менеджер-архитектор требований к программному продукту
Требования к версии программного продукта и комплекту документации	Требования к модификации и улучшению функций и качества версии программного продукта	Менеджер-архитектор требований к программному продукту	Менеджер и заказчик проекта программного продукта
Версии требований заказчика и пользователей к программному продукту	Адаптация требований к характеристикам внешней среды пользователей	Заказчик или менеджер сопровождающий версию программного продукта пользователей	Менеджер, сопровождающий версию программного продукта пользователей

Процессы УК включают работы по идентификации конфигурации, контролю изменений, определению базовой версии разработки и архивированию программного продукта, включая соответствующие документы жизненного цикла, по аудиту конфигурации, компоновке и поставке программного продукта в течение всего жизненного цикла системы. Процессы УК, выполняемые совместно с другими процессами жизненного цикла, направлены на достижения ***следующих основных целей*** [34]:

- обеспечить возможность оценки соответствия требованиям заказчика результатов жизненного цикла программного комплекса;
- обеспечить определяемую и управляемую конфигурацию КП на протяжении всего жизненного цикла;
- обеспечить контрольные точки для проверки, оценки состояния и контроля изменений посредством управления элементами конфигурации и определения базовой версии программного продукта;
- обеспечить контроль над тем, чтобы фиксировались дефекты и ошибки, а изменения регистрировались, утверждались и реализовались;
- гарантировать надежное физическое архивирование, восстановление и сопровождение единиц конфигурации и документов программного продукта.

Управление конфигурацией должно быть организовано таким образом, чтобы можно было гарантировать беспристрастность и независимость специалистов при достижения целей управления конфигурацией. Для того чтобы управление конфигурацией было эффективным, следует определить ***организационную структуру коллектива специалистов***. Структура должна включать функцию управления конфигурацией; взаимодействующие организации; службы проектирования, закупок и контрактов; управление данными, изготовление, обеспечение качества и другие дисциплины, которые могут быть привлечены, охватывая, если необходимо, субподрядчиков и поставщиков. В рамках организации проекта следует идентифицировать инстанцию, уполномоченную утверждать конфигурационные базы и любые изменения к ним, обычно это ***совет по конфигурации***. Руководитель проекта ***может учредить совет по конфигурации***, который будет иметь полномочия анализировать и утверждать или не утверждать программу и процедуры управления конфигурацией, выбор объектов конфигурации, конфигурационные базы и изменения к этим базам, включая отклонения и разрешения на отклонение.

Конфигурационная идентификация включает методы и средства, с помощью которых можно однозначно устанавливать и различать версии КП, входящие в них компоненты (единицы конфигурации – ЕК), их варианты и модификации, а также родословное дерево объектов конфигурации, технических требований и идентифицированный комплект документации. Кроме того, каждый вариант модуля, компонента или комплекса и их изменений должен соответствовать правилам нумерации (идентификации). Цель идентификации конфигурации заключается в ***однозначной маркировке*** каждой единицы конфигурации. В конкретном проекте следует разработать ***правила нумерации ЕК*** и применять их для идентификации объектов конфигурации, документов по конфигурации и изменений. Идентификационные номера должны быть уникальными.

Конфигурационный учет составляют методы и средства регистрации и отслеживания состояния объектов – единиц конфигурации, накопления и классификации отчетов о всех реализованных и отвергнутых изменениях вариантов компонентов и КП в целом. Совокупность отчетов должна обеспечивать идентификацию изменений и однозначное отражение текущего состояния КП и его компонентов, а также накопление историй последовательных модификаций ЕК, приведших к данному состоянию их структуры, функций и характеристик. В определенные моменты нужно фиксировать версии всех объектов и компонентов, составляющих программный продукт или систему.

Управление запросами на изменения включает регистрацию, отслеживание и анализ запросов на модификацию КП, компонентов и данных от субъектов, взаимодействующих с системой (см. рис. 2.14). Оно включает процессы принятия решений для планирования необходимых изменений и процессы их реализации. Управление запросами на изменения представляет собой ***центральную часть Концепции управления конфигурацией***. Хотя с запросами на доработку и дефектами связана достаточно похожая информация, в процессах УК они часто обрабатываются совершенно по-разному. Обычно в определении этого процесса участвуют различные подразделения (руководители проектов, разработчики, специалисты по тестированию). Внешний запрос способен в свою очередь породить несколько внутренних технических запросов на доработку. Следующий шаг после установления типов контролируемых запросов, состоит в уточнении объема информации, фиксируемой в ходе жизненного цикла запроса.

Представление запроса на изменение предполагает его регистрацию. Запросы на доработку могут поступать из различных источников. Основные данные, регистрируемые при возникновении таких запросов, это важность требуемых изменений для заказчика или пользователей, подробности запроса и личность инициатора. Иногда запрос на доработку появляется в самой организации в процессе производства, тестирования или внутреннего использования проекта. Необходимо также учесть возможные последствия доработки – изменение доли рынка, увеличение прибыли, влияние на продажи и поддержку пользователей.

Изменения программ и/или данных модулей и программных компонентов подвергается наименее формализованному и защищенному от случайностей конфигурационному управлению. Их изменения в процессе тестирования обычно не требуют санкции руководителей проекта. Версии функциональных групп программ и комплексов программ в целом могут корректироваться только с разрешения руководителей соответствующего ранга. Тем самым должны предотвращаться несогласованные и несанкционированные изменения, способные снизить качество и целостность версий программного продукта. Изменения этих версий допускаются пользователями или поставщиками в пределах, ограниченных эксплуатационными документами по адаптации к характеристикам и особенностям внешней среды и условий применения конкретной версии КП. Правила модификации в проекте должны определять, к каким компонентам разработчики различных уровней будут иметь доступ, указывать, доступен определенный компонент в данный момент только для чтения или для чтения и модификации.

Систему корректирующих действий следует реализовать для обработки каждого дефекта, обнаруженного при модификации, находящихся под контролем конфигурации (см. рис. 2.14). Необходимо гарантировать, что все обнаруженные дефекты немедленно регистрируются и вводятся в систему УК, необходимые действия инициируются, принятые решения осуществляются, состояние корректирующих действий прослеживается, сообщения о дефектах и модификациях сопровождаются в течение всего срока действия контракта. Каждый дефект и модификация должны быть классифицированы по категориям и приоритетам. Корректирующие действия должны быть оценены, чтобы определить, были ли дефекты устранены, а изменения были правильно выполнены без внесения дополнительных дефектов.

Решение о корректировке конфигурации КП состоит в выборе специалистами: выполнить запрос на изменения, отложить реализацию или отклонить запрос. Для запроса на доработку решение обычно выносит руководитель проекта или аналитик. Затем относительно каждого из них принимается решение о реализации в данной версии продукта, отсрочке или отклонении. Процесс принятия решения для дефектов отличается и зависит от двух факторов: текущей фазы цикла разработки и необходимых для реализации усилий. На заключительных стадиях проекта целесообразно вводить формальный процесс рассмотрения и внесения исправлений, он должен быть направлен на ограничение изменений и допуск только самых критических исправлений на стадиях стабилизации кода и квалификационного тестирования.

Реализация корректировок при запросе на доработку может требовать проведения дополнительных проектных работ, поскольку в систему добавляются новые функции. Для исправления дефекта более важно воссоздать среду, где он проявляется и где, затем будут тестироваться сделанные модификации. В случае запроса на доработку это означает внесение в документацию описания новых функций; в случае дефектов – исправление документации, если устраненная ранее ошибка изменила поведение пользователя при взаимодействии с системой.

Редакциям документов ЕК присваиваются определенные ***статусы, определяющие их качество и пригодность для различных операций***. В каждом проекте один из статусов является рекомендуемым, базовым. Последняя редакция ЕК в интеграционном потоке проекта, получившая такой статус, по умолчанию предлагается разработчикам при выполнении операций – ***обновить***. Если редакция ЕК вызывает ошибки при сборке, ей необходимо присвоить ***статус – отклонена***, что позволит избежать использования данной редакции в дальнейшей работе.

Контроль корректности конфигураций версий компонентов, КП и данных предназначен для систематической оценки предполагаемых изменений ЕК и координированной их реализации с учетом соответствия спецификациям и требованиям заказчика, эффективности каждого из них и затрат на выполнение изменения. Цель контроля изменений – обеспечить регистрацию, оценку, рассмотрение и утверждение корректировок на протяжении всего жизненного цикла КП. Контроль изменений должен гарантировать, что каждое измене-

ние единицы конфигурации учтено в изменении идентификации конфигурации КП.

Работы ***по просмотру и прослеживанию корректности изменений*** должны сопровождать реализацию и контроль изменений ЕК. Целью является оценка дефектов и модификаций, их утверждения, реализации утвержденных изменений и обратной связи к процессам, на которые изменение воздействует, путем использования методов контроля изменений. В интеграционный поток следует отправлять только те модификации, которые предшествуют операции регистрации ЕК. Перед сборкой базовой версии комплекса программ рекомендуется ***блокировка интеграционного потока и запрет изменений*** ЕК. В результате интегратор может проводить сборку и создавать базовую редакцию КП для стабильного множества элементов исходного кода ЕК.

Сборка версии программного средства – первый уровень тестирования, проводимый интегратором, чтобы убедиться в том, что все модифицированные файлы ЕК могут быть собраны в единый компонент или комплекс программ (см. рис. 2.14). Если сборка прошла успешно, то следует перейти к квалификационному тестированию или повысить статусы редакций компонентов ЕК. В некоторых случаях интегратор может самостоятельно внести исправления и запустить сборку повторно. Сложные проблемы могут возникать, когда модификации ЕК отправлены несколькими сотрудниками. Чтобы разобраться в ошибочной ситуации, интегратор вправе привлечь разработчиков изменений, они внесут дополнительные изменения в ЕК и снова отправят их в поток интеграции. Чтобы дать им возможность отправить изменения повторно, интегратору необходимо временно исключить их из блокировки интеграционного потока.

Сформированные базовые версии единиц конфигурации должны регистрироваться в контролируемых библиотеках КП и позволять ссылаться, управлять и прослеживать их изменения. Они должны быть защищены от внесения любых несанкционированных изменений. Конфигурационная база состоит из всех утвержденных документов, которые определяют программную продукцию или компоненты в данный момент. Её следует устанавливать всегда, когда это необходимо для определения эталонной конфигурации КП и/или компонентов в течение их жизненного цикла, который служит отправной точкой для последующей деятельности.

После проведения работ по реализации и контролю совокупности изменений должна быть разработана и зафиксирована очередная базовая версия ЕК, производная от ранее установленной базовой версии. ***Цель установления очередной базовой версии*** – определить основу для последующих работ процессов жизненного цикла КП и позволить осуществлять ссылки, управлять и прослеживать единицы конфигурации. Базовые версии следует хранить в контролируемых библиотеках, обеспечить их целостность, они должны быть защищены от внесения несанкционированных изменений.

Квалификационные базовые тесты предназначены для проверки корректности функционирования готового программного продукта. Исполнение базовых тестов позволяет быть уверенным в корректности заказного ***программного продукта***. Если тестирование прошло успешно, значит, интегрированные изменения достаточно стабильны, статусы новых редакций ЕК можно повысить до рекомендованного базового уровня, и специалистам разрешено использовать их при обновлении своего рабочего пространства. После завершения сборки и создания базовой редакций программного продукта интеграционный поток разблокируется. Специалистам снова разрешается отправлять изменения в версии КП и обновлять рабочие версии файлов ЕК новыми редакциями.

Составление отчетов о состоянии конфигурации должно предоставлять заказчику информацию о идентификации конфигурации и всех отклонениях от зарегистрированных конфигурационных баз. Проверки конфигурации следует проводить до принятия и утверждения конфигурационной базы с целью гарантии того, что программный продукт соответствует ***контрактным требованиям*** спецификаций, и что он точно отражен в документах по конфигурации. Проверка конфигурации может потребоваться для официальной приемки заказчиком комплекса программ. Существуют, как правило, ***два типа проверок документов конфигурации***:

- проверка функциональной конфигурации: официальная экспертиза с целью установления того, что единицы конфигурации имеют такие эксплуатационные и функциональные характеристики, какие требуются в документах требований по данной конфигурации;
- проверка физической конфигурации: официальная экспертиза конфигурации непосредственно после сборки и изготовления комплекса программ с целью утверждения того, что он соответствует конфигурационным документам на продукцию.

Архивирование и тиражирование базовых версий программных продуктов и документов должны гарантировать использование исключительно санкционированных версий программного продукта, так что официальные версии могут быть получены только из архива. Цель работ по архивированию и применению документов – обеспечить получение документов жизненного цикла КП, для копирования, повторной генерации, повторного тестирования и модификации программного продукта. Должны быть регламентированы ***полномочия специалистов*** по выпуску базовых версий единиц конфигурации заказчику:

- документы жизненного цикла, связанные с программным продуктом должны быть получены заказчиком из утвержденного источника от организации-разработчика;
- процесс копирования и тиражирования должен быть верифицирован, чтобы гарантировать получение точных копий, и должны существовать процедуры, гарантирующие безошибочное копирование исполняемого объектного кода программного продукта;
- должны быть установлены полномочия по выпуску базовых версий единиц конфигурации и компонентов программного продукта, загружаемого в вычислительную систему или оборудование;
- необходимо установить процедуры хранения, включения, удаления компонентов конфигурации, чтобы удовлетворить требования пригодности к применению и обеспечить возможность модификации версии программного продукта.

Этапы и процедуры при управлении конфигурацией заказных программных комплексов

Конфигурационное управление в значительной степени обеспеченно, если ***КП имеет четкую структуру***, а его компоненты – унифицированные интерфейсы по управлению и информации. Для этого правила модульно-иерархического построения КП должны детализироваться, до уровня конкретных методик создания и оформления модулей, компонентов и межмодульного взаимодействия. При этом унифицированные межмодульные интерфейсы целесообразно, по возможности, упрощать и ослаблять, а также подготавливать условия для их проверок при реальном функционировании программ. Проектирование КП сверху вниз и последующее сохранение четкой структуры интерфейсов значительно облегчают управление конфигурацией и продлевают срок жизни версий комплексов программ. Регистра-

ция и учет истории этого процесса обеспечивает возможность его контроля и пошагового восстановления выполненных изменений (отката) при ***выявлении вторичных дефектов***, внесенных в процессе разработки модификаций для очередной базовой версии программного продукта.

В результате развития сложного комплекса программ в архиве тиражирования и обеспечения сохранности образуется ***зона сопровождения*** – комплект конфигураций, доступных для изменений базовых версий программного продукта. Число таких сопровождаемых ***базовых версий разработчика конфигураций*** или глубина сопровождения практически всегда не менее двух версий и редко превышают четыре версии. Для крупномасштабных заказных КП это соответствует рациональному времени жизни и тиражирования каждой очередной базовой версии программного продукта около 1 – 2 лет.

В схеме основных этапов, регламентирующих конфигурационное управление КП, представлен ***комплекс процедур и состав соответствующих хранилищ информации***, организующих эти процессы – рис. 2.15. Для реализации на практике приведенных выше концепций и процедур, требований и планов сопровождения и управления конфигурацией сложных заказных программных комплексов необходимы организационные мероприятия, гарантирующие участникам производства ***определенную культуру, дисциплину разработки и выполнения модификаций***. Такая ***организационная система*** должна обеспечивать специалистам разной квалификации и роли в производстве, возможность взаимодействия при решении требуемых комплексных задач, для накопления, хранения и обмена упорядоченной информацией о состоянии и изменениях компонентов и продукта.

Для общения с заказчиком и пользователями и накопления информации о выявляемых недостатках в тиражируемых сложных КП целесообразно выделение группы аналитиков высокой квалификации, овладевших всеми функциональными возможностями данного программного продукта [13, 19, 34].

Группа предварительной селекции – аналитиков предполагаемых изменений, должна иметь в своем зарегистрированном и аннотированном арсенале практически весь комплекс тестов, применявшихся при испытаниях опытного образца и предыдущих версий КП для возможности регрессионного тестирования.

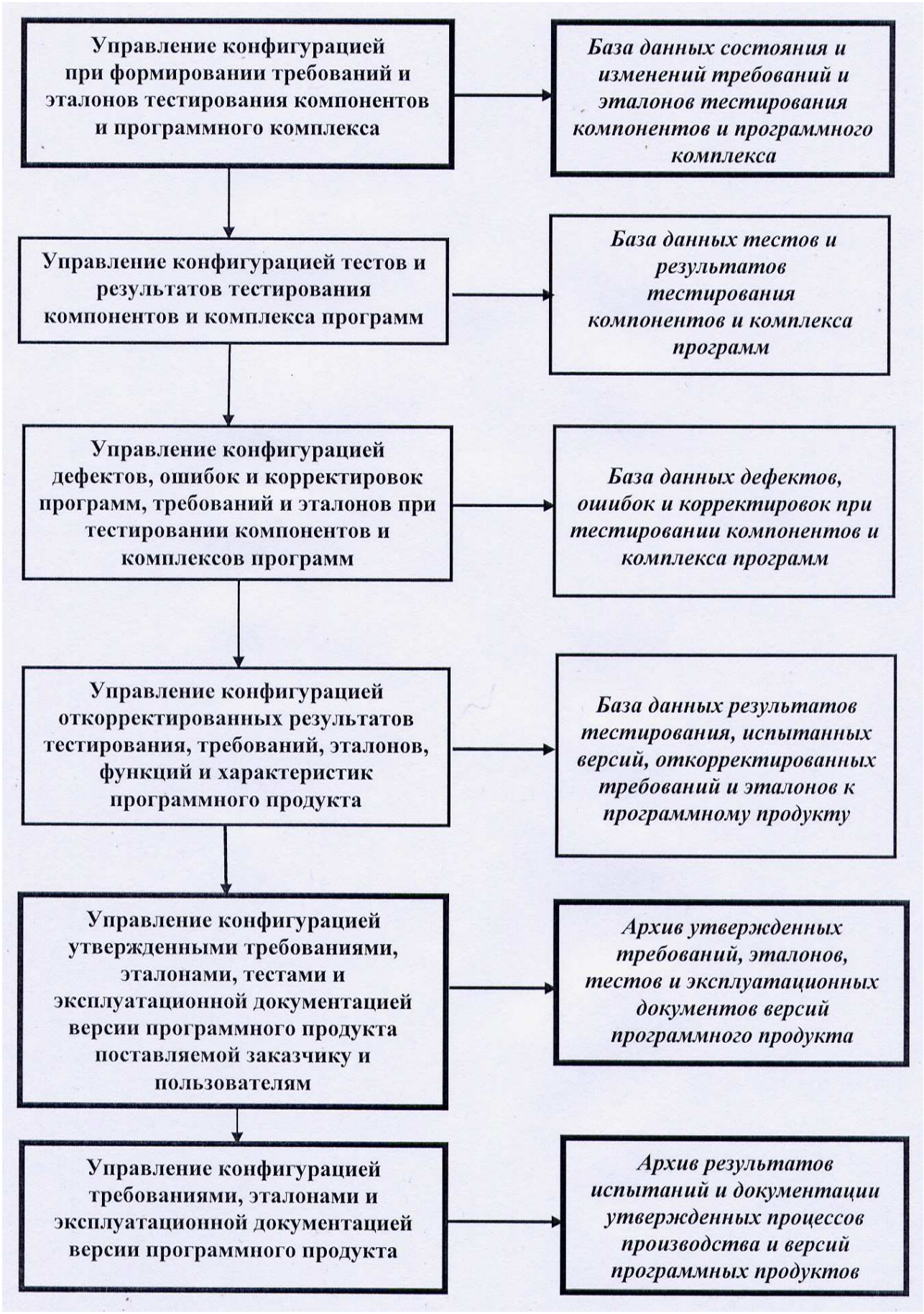
Управление конфигурацией при формировании требований и эталонов тестирования компонентов и программного комплекса
База данных состояния и изменений требований и эталонов тестирования компонентов и программного комплекса
Управление конфигурацией тестов и результатов тестирования компонентов и комплекса программ
База данных тестов и результатов тестирования компонентов и комплекса программ
Управление конфигурацией дефектов, ошибок и корректировок программ, требований и эталонов при тестировании компонентов и комплексов программ
База данных дефектов, ошибок и корректировок при тестировании компонентов и комплекса программ
Управление конфигурацией откорректированных результатов тестирования, требований, эталонов, функций и характеристик программного продукта
База данных результатов тестирования, испытанных версий, откорректированных требований и эталонов к программному продукту
Управление конфигурацией утвержденными требованиями, эталонами, тестами и эксплуатационной документацией версии программного продукта поставляемой заказчику и пользователям
Архив утвержденных требований, эталонов, тестов и эксплуатационных документов версий программного продукта
Управление конфигурацией требованиями, эталонами и эксплуатационной документацией версии программного продукта
Архив результатов испытаний и документации утвержденных процессов производства и версий программных продуктов

Рис. 2.15

Тесты накапливаются, упорядочиваются и каталогизируются в базе данных тестирования. Эти же специалисты осуществляют развитие набора тестов для подтверждения наличия и локализации частных дефектов и ошибок, а также для первичной оценки целесообразности реализации предложений по развитию и модернизации программ.

Для ***установления достоверности*** сообщений пользователей о выявленных дефектах и ошибках необходима детальная регистрация сценариев и условий, при которых проявляются аномалии. Установление разработчиками достоверности ошибок начинается с тестирования эталонной базовой версии КП при исходных данных пользователя, обнаружившего дефект. Если проявляется ошибка, то она регистрируется как подтвержденная при зафиксированных тестовых данных. При отсутствии проявления ошибок на эталонной версии, при тех же исходных данных целесообразно проводить последующее тестирование копии версии, адаптированной к условиям применения у пользователя. Если и при этом ошибка не проявляется, то регистрируется её не подтверждение, о чем сообщается пользователю, и информация о дефекте снимается с учета.

Идеи небольших корректировок программ целесообразно накапливать отдельно от предложений по существенному совершенствованию системы. Таким образом, последовательно формируется документ – ***описание выявленных дефектов и предложений по корректировке КП и компонентов***, содержащий исходные данные для планирования доработок в процессе сопровождения. Для повышения качества новых версий ***руководитель и совет конфигурационного управления*** анализируют все предлагаемые изменения и выделяют из них целесообразные для анализа возможного использования в очередной версии (см. рис. 2.15). Селекция проводимых изменений в версиях сложных КП требует формализации анализа этого процесса и документирования предполагаемых и утвержденных изменений.

В совете конфигурационного управления сосредоточивается информация для планирования основных операций по доработке и выпуску очередных версий КП. Специалисты должны оценивать степень срочности исправления ошибок и проведения модернизаций, а также выявлять условия, позволяющие достаточно полноценно эксплуатировать программы до выполнения всех запланированных изменений при выпуске очередной версии.

Для принятых к внедрению изменений разрабатывается ***план доработок программ*** и определяется конкретный специалист, ответст-

венный за каждую корректировку программы. Изменения программ могут потребовать либо полной замены модуля или группы программ, либо небольшого изменения текста программного модуля, описания данных или констант. При полной замене компонентов они подлежат тестированию, как все вновь созданные компоненты. Группы изменений регистрируются в ***описании и базе данных подготовленных корректировок*** для последующего использования с указанием специалистов их подготовивших и лиц, принявших решение о реализации. Подготовленные и утвержденные предложения на изменения программного продукта поступают к специалистам, осуществляющим планирование их внесения в реальные программы, а также разработку и корректировку конкретных компонентов и документов. После выполнения доработок разработчиками компонентов и корректировки документации, следует окончательная проверка корректности изменений, которая осуществляется ***специалистами по квалификационному тестированию и испытаниям очередной версии программного продукта*** (см. рис. 2.15).

Тестирование необходимо сосредоточивать на компонентах впервые вводимых или значительно модифицируемых в данной версии. Если изменения в программе или в данных невелики, то тестирование обычно можно ограничить компонентами, непосредственно связанными с выполненной корректировкой. При этом следует учитывать, что корректировки программ в 10 – 30% случаев могут содержать ошибки и требуют тщательного тестирования не только тех частей, где внесены изменения.

Наличие в сложных программах глубоких межмодульных связей по управлению и по информации вызывают необходимость частичного тестирования тех компонентов и их интерфейсов, где по первому впечатлению корректировки не оказывают влияния. Поэтому ***регрессионному тестированию*** необходимо подвергать также некоторые части и интерфейсы комплекса программ, которые не подвергались изменениям. Для этого могут использоваться тесты, ранее применявшиеся при испытаниях предыдущей версии. Проверенные таким образом изменения регистрируются в ***описании и базе данных утвержденных и проведенных корректировок для следующей базовой версии программного продукта или для срочных извещений пользователям***.

Конфигурационная база заказного программного продукта (см. рис. 2.15) должна быть зарегистрирована официально в опреде-

ленный момент времени и использоваться в качестве отправной точки для контроля за состоянием конфигурации и утвержденными изменениями очередной версии.

Пользователям может поставляться копия базовой версии для самостоятельной ***адаптации*** программного продукта к характеристикам внешней среды и особенностям конкретного применения. В этом случае адаптация должна проводиться строго по инструкциям разработчика этой версии и должны быть запрещены любые дополнительные изменения за пределами, предписанными документацией. Подобные изменения автоматически снимают гарантии разработчика, и ответственность за корректность адаптации возлагается на пользователя. Для независимого удостоверения достигнутого качества проведенных модификаций в испытанной и утвержденной заказчиком версии программного продукта по заявке разработчика, заказчика или пользователей она может подвергаться обязательной или добровольной ***сертификации*** (см. главу 2.8).

Наличие срочных исправлений ошибок и запросов пользователей на частичные изменения в эксплуатируемых версиях, а также ряд других причин могут приводить к появлению между базовыми версиями КП ряда промежуточных пользовательских версий. Для этого, наряду с выпуском очередных базовых версий, приходится выпускать ***временные извещения на исправления выявленных ошибок и на корректировки,*** а также извещения об основных изменениях функций, которые проведены в новой версии, по сравнению с предыдущей.

Для эффективной реализации перечисленных процессов представленная схема организации конфигурационного управления должна быть поддержана ***службой и базой данных тиражирования, архивирования и гарантированного хранения*** всей необходимой информации о компонентах, их корректировках, версиях, авторах и их правах на изменения. Эта служба должна обеспечивать поставку пользователям только утвержденных копий версий КП и/или их компонентов с полным, адекватным комплектом эксплуатационных документов, а также извещений на частные изменения конкретных, ранее приобретенных версий. Для гарантированного сохранения состояния и результатов модификаций программ и обеспечения возможности их анализа на любой стадии проекта, а также «***отката» по истории выполненных корректировок***, следует организовать четкую систему хранения и копирования подлинников, дубликатов и копий одного и того же программного продукта и документов. Эта сис-

тема должна иметь соответствующих руководителей и специалистов, ответственных за полную сохранность всех результатов истории изменений конфигурации конкретного программного продукта в течение заданного интервала времени. Для гарантированного сохранения от случайного или преднамеренного разрушения базовой версии КП в составе коллектива конфигурационного управления должны быть выделены специалисты, ответственные за сохранность подлинников физических носителей и документации испытанных и утвержденных версий программного продукта.

Внедрение заказного программного продукта через специализированный архив предполагает широкие масштабы и промышленный подход к тиражированию документации и программных комплексов, к поставке, обслуживанию, управлению конфигурацией и сопровождению. Передача программного продукта в архив завершается ***приемо-сдаточными испытаниями***, которым предшествуют опытная эксплуатация и испытания разработчика.

В каждом производстве заказных программных продуктов должен быть организован регламентированный процесс управления конфигурацией и сопровождения, обеспечивающий для коллектива специалистов единую ***среду производства, хранения, изменения и утверждения модификаций***, адекватных реальному содержанию объектного кода программ, текстовых данных и тестов программного продукта. Процесс организации и технологического обеспечения УК должен быть ориентирован на слаженную коллективную работу различных профессионалов, объединенных единой целью развития и модификации требуемого заказчиком комплекса программ с заданными функциями и высоким качеством. Каждый участник проекта в соответствии со своими функциональными обязанностями должен иметь доступ к необходимой для него ***корректной информации***, и ограничен возможностями обращения к несанкционированным для него данным.

Организация документирования заказных программных комплексов

Технологической основой управления конфигурацией КП являются ***системы управления базами данных*** (СУБД), адекватные целям и функциям проектов, структурированные по назначению и содержанию данных в выделенных подсистемах хранения (см. рис. 2.15 – правая колонка). Они должны обеспечивать возможность

управления организационной и проектной деятельностью коллективов специалистов, универсальное хранилище в них необходимых данных с инструментами наполнения, корректировки, поиска и контроля информации, соответствующей их профессиональной деятельности. Должны быть упорядочены деловые коммуникации между специалистами разных категорий, управление динамическими процессами выполнения изменений и транспортировки корректировок между подсистемами в соответствии с целями их использования специалистами.

Первоначально должен быть разработан ***проект архитектуры системы технологического обеспечения управления конфигурацией, а также Руководство по её применению***, настроена выбранная СУБД на управление основными взаимодействующими подсистемами базы данных с учетом класса и масштаба предполагаемого проекта. По мере развития жизненного цикла производства комплекса программ подсистемы базы данных УК должны поэтапно заполняться документами от заказчика и разработчиков соответствующих квалификаций, и контролироваться менеджерами проекта.

Эта ***информация в подсистемах базы данных УК*** должна быть защищены от случайных и преднамеренных искажений путем организованного санкционирования, дублирования и контроля модификаций, истории их создания и изменения в процессах жизненного цикла КП. Необходимо гарантировать сохранность версий изменений с учетом их важности для результатов всего проекта. Особенно защищенным от искажений и разрушения следует сохранять ***архив базовых версий программных продуктов и документов***, прошедших успешные испытания, утвержденных заказчиком [34].

В соответствии с основными задачами специалистов производства на рис. 2.15 представлен ***вариант организации*** частных подсистем УК базы данных информационного обеспечения модификаций, ***ориентированные на определенные процессы*** и компоненты ЖЦ комплексов программ. Для каждой подсистемы целесообразно выделять достаточно автономную базу данных компонентов УК с ограниченным доступом только для определенных категорий специалистов. Эти фрагменты базы данных могут быть построены на стандартизированной основе СУБД производства, взаимодействовать с аналогичными по структуре предшествующей и последующей базами данных. Для каждого сложного проекта комплекса программ целесообразно оформлять и утверждать ***Руководство и схему базы данных, обеспе-***

чивающей документооборот и управление конфигурацией, а также категории ответственных лиц за их поэтапную реализацию, контроль и сохранность информации. Для этого должны быть выделены ***руководители и коллективы специалистов, которые должны планировать, утверждать, выпускать, распространять и сопровождать комплекты документов***. Они должны стимулировать разработчиков программных комплексов, компонентов и их изменений, осуществлять непрерывное, регламентированное документирование процессов и результатов своей деятельности, а также контролировать полноту и качество отчетных документов.

Структура документации и формы отдельных документов, используемых для конфигурационного управления программами, должны позволять точно документально описывать и идентифицировать каждую оформленную версию программных компонентов и продуктов в целом в любое время на всем протяжении их жизненного цикла. Особое значение при управлении конфигурацией имеет ***документация на реализованные изменения и тесты***, с помощью которых проверяется корректность версий компонентов и комплекса в целом. Эта документация должна позволять восстанавливать ***историю разработки и проверки*** каждого изменения любого компонента. На базе всего комплекса использованных тестов создается и документируется для каждой версии программного продукта эталонная тестовая (контрольная) задача и контрольные результаты её решения. Производство и тестирование изменений компонентов и КП всегда несколько опережают их документальное оформление. В течение этого времени возможны отдельные уточнения изменений в версиях. В результате документация должна непрерывно «***догонять***» реальное состояние программного продукта. Для упорядочения этого процесса стандартами установлена возможность оперативного выпуска ***предварительных официальных извещений на частные изменения***. Эти извещения регистрируются как временные и погашаются при полном оформлении документации на очередную версию программного продукта и все изменения.

Управление документацией должно непрерывно поддерживать её полноту, ***корректность и согласованность с программным продуктом***. Адекватность документации требованиям, состоянию текстов и объектных кодов заказных программ должна инспектироваться и удостоверяться (подписываться) ответственными руководителями и заказчиками проекта. ***Ошибки и дефекты документов не менее***

опасны для применения КП, чем ошибки в структуре, интерфейсах, файлах текстов программ и в содержании данных. Поэтому к разработке, полноте, корректности и качеству документации необходимо столь же тщательное отношение, как к производству и изменениям текстов программ и данных.

Реализация документов в значительной степени определяет достигаемое качество сложных заказных программных продуктов, трудоемкость и длительность их создания. Для этого должна формироваться и использоваться ***регламентированная стратегия, стандарты, распределение ресурсов и планы создания, изменения и применения документов*** на программы и данные сложных систем. В общем случае должны быть ***выделены руководители и коллектив специалистов***, которые будут планировать, описывать, утверждать, выпускать, распространять и сопровождать комплекты документов. Официальная описанная и утвержденная стратегия документирования должна ***устанавливать дисциплину***, необходимую для эффективного создания высококачественных документов на продукты и процессы в жизненном цикле КП.

При создании особо сложных систем целесообразно выделение специального коллектива, обеспечивающего организацию и реализацию основных ***системных работ по документообороту***. Организация процессов документирования должна обеспечивать гибкое и точное изменение документов – ***сопровождение и конфигурационное управление версиями и редакциями каждого документа***. Эти процессы и поддерживающие их средства автоматизации должны быть адекватными тем, которые применяются для непосредственных объектов производства – комплекса программ и данных. Для хранения, тиражирования и распространения документов, сложных КП высокого качества, следует выделять группу специалистов, ответственных за ***контроль, обеспечение и гарантированное сохранение документации***. Для критических заказных систем документация на программы и данные должна храниться и дублироваться на различных типах носителей [7, 13, 29].

Формированию требований к комплексу программ должно сопутствовать ***создание требований, отражающих его документооборот***, вследствие чего эти процессы во многом подобны. Масштаб проекта и спецификация требований к КП непосредственно отражаются на составе, содержании и объеме документации, необходимой различным участникам проекта. Разработчикам необходимо выработ-

тать профессиональные приемы для ***понимания и изложения в документах потребностей заказчиков и пользователей***, управления масштабом проекта, построением системы и документации программного продукта.

В зависимости от сложности проекта окончательным результатом работ должны быть детализированные и утвержденные документы спецификаций к номенклатуре, свойствам, значениям качества и документации производства, которые достаточны для его полноценного рабочего производства и последующей, эффективной эксплуатации. Эти ***требования к документам должны закрепляться в контракте и техническом задании***, по которым разработчик впоследствии должен отчитываться перед заказчиком при завершении проекта (см. рис. 2.15).

Требования к функциональным характеристикам, качеству, составу и содержанию документов, утвержденные после предварительного проектирования, могут быть закреплены в техническом задании ***как обязательные для детального и рабочего проектирования***. Эти данные могут использоваться при последующем оценивании качества документации при их сопоставлении с требованиями в процессе квалификационных испытаний или сертификации программного продукта. Однако для сложных заказных продуктов может потребоваться уточнение требований к качеству документации при детальном проектировании.

Номенклатура ***технологических документов*** в жизненном цикле крупного комплекса программ может доходить ***до 30 – 50 видов***, среди которых наибольшее влияние на объем документации оказывают: спецификации программ и данных, тексты программ с комментариями, тестовые сценарии и результаты тестирования компонентов и модулей. Для отражения совокупности этих документов в среднем на каждую строку текста программы может требоваться от 10% до полной страницы документации. Таким образом, технологическая документация для всего жизненного цикла КП размером 10^6 строк может составить около ста тысяч страниц или ста томов по тысяче страниц. Вряд ли целесообразно изготавливать такой объем твердых копий документов на бумаге. Большая часть этих документов ***может оставаться в файлах*** (сотни мегабайт), однако каждый документ должен быть оформлен в соответствии со стандартами и шаблонами, и скреплен подписями разработчиков и, где нужно, заказчика или ру-

ководителя. Изменение этих документов должно санкционироваться так же, как твердых копий.

Приведенные оценки следует рассматривать ***только как ориентиры***, которые в реальных проектах могут изменяться на порядок в ту или иную сторону в зависимости от требований заказчика и характеристик проекта. Оценки объема предстоящей разработки технологической и эксплуатационной документации целесообразно проводить на этапе детального проектирования с учетом реальных характеристик КП, что позволит избежать неприятных сюрпризов, вызванных превышением затрат на реализацию документов.

Для реализации качественного документирования должны быть созданы регламентирующие документы, охватывающие обязанности и ответственность специалистов за качество конкретных документов. Процессы, работы, шаблоны и задачи ЖЦ должны селектироваться и включать в себя перечень документов, которые ***обязательно нужно разработать и информацию о персональной ответственности за них***. Все исходные данные и решения по адаптации номенклатуры, структуры и содержания документов должны быть документированы и утверждены руководством проекта вместе с обоснованием их целесообразности.

Для реализации на практике требований и планов документирования в жизненном цикле программных средств необходимы ***организационные мероприятия***, гарантирующие участникам проектов ***определенную культуру, дисциплину разработки и применения документов***. Такая организационная система должна обеспечивать специалистам разной квалификации и специализации возможность взаимодействия при решении требуемых комплексных задач, для накопления, хранения и обмена упорядоченной информацией о состоянии и изменениях компонентов проекта.

Подготовка эксплуатационной документации для заказных программных продуктов

В состав версий заказного программного продукта входят ***подсистемы документооборота и комплекты документов пользователей***. Эти комплекты и содержание документов следует адаптировать в соответствии с требованиями и характеристиками проекта. В системах реального времени в ряде случаев могут отсутствовать документы административного управления, а также сокращены в шаблонах требования и функции оперативных пользователей. В комплек-

сах программ административных систем может доминировать документооборот администраторов, действующих совместно с оперативными пользователями при реализации основных функций программного продукта. В некоторых автоматизированных системах реального времени программный продукт является органическим компонентом системы управления и внешней среды, и эксплуатационная документация должна отражать в основном контрольные функции, инсталляцию и оценки целостности функционирования программного продукта в системе.

Эксплуатационная документация должна обеспечивать ***эффективное применение программного комплекса*** и точно отражать его назначение, функции, характеристики и требования для использования квалифицированными специалистами-пользователями. Для этого эксплуатационные документы необходимо тестировать на полное соответствие выполнения всей ***совокупности требований на программный комплекс***, согласованных между заказчиком и разработчиками. В результате эти документы можно использовать как отдельный при производстве ***третий эталон и вид тестов пользователей*** корректной реализации требований к функциям и характеристикам программного продукта. Практическое апробирование документов при применении комплекса программ является практическим методом тестирования корректности реализации требований к программному продукту, завершающим полный цикл контроля его качества. Разработчики документов должны обеспечивать комфортное и корректное применение комплекса программ пользователями на основе ясного и непротиворечивого изложения в документах технологических процедур и операций для его штатного функционирования и получения ***требуемых результатов*** [7, 19, 29].

Организация документирования должна определять стратегию, стандарты, процедуры, распределение ресурсов и планы создания, изменения и применения документов на комплекс программ. Для этого в общем случае должны быть выделены ***специалисты***, которые обязаны планировать, утверждать, выпускать, распространять и сопровождать комплекты апробированных и утвержденных эксплуатационных документов. Они должны стимулировать разработчиков программных комплексов осуществлять непрерывное, регламентированное документирование процессов и результатов своей деятельности, а также контролировать полноту и качество утвержденных эксплуатационных и отчетных документов.

Состав и содержание комплекта эксплуатационных документов конкретного программного комплекса целесообразно адаптировать разработчикам к его особенностям и свойствам ***на основе использования стандартов и типовых структур – шаблонов*** для двух классов продуктов, которые в наибольшей степени различаются особенностями эксплуатации. ***Первый класс*** составляют сложные ***заказные программные комплексы*** автоматизированного управления динамическими объектами и процессами в реальном масштабе времени. В процессе их применения допускается минимальное вмешательство пользователями в процедуры управления и необходим, соответственно, небольшой объем эксплуатационных документов, выделяемых из стандартизированного полного комплекта документов программного продукта. Для программных комплексов ***второго класса*** возможно применение пользователями более широкого набора процедур управления, которые должны быть регламентированы достаточно полным набором и подробным содержанием документов. Пользователей таких программных комплексов можно разделить на две крупных группы, каждая из которых должна быть обеспечена комплектной эксплуатационной документацией:

- администраторы, подготавливающие программный комплекс к эксплуатации, обеспечивающие их функционирование, контроль и использование по прямому назначению;
- операторы – пользователи, реализующие применение программных комплексов в системе, их функционирование, обработку и анализ результатов.

Документация администрирования при эксплуатации системы должна обеспечивать поддержку первичной инсталляции, безопасного функционирования и восстановления программ и данных после сбоев. Администратор системы и программного комплекса должен быть информирован о всех изменениях функционирования устройств системы и внешней среды, могущих привести к сбою или возникновению аварийной ситуации, и предпринимать соответствующие профилактические действия. Для этого требуется полная информация о требованиях к программному комплексу, к компонентам системы и внешней среды, которые имеют свои особенности в управлении с помощью специальных программ, поддерживающих администрирование и управление системой.

Документация для оперативных пользователей программных комплексов в системе, их функционирования, обработки и анализа результатов должна обеспечивать оперативное взаимодействие поль-

зователей с различными аппаратно-программными реализациями терминалов. Для этого необходима унификация концепции, архитектуры, функций и методов визуализации пользовательского интерфейса. Типовые ***формы-шаблоны документов*** и процедуры работы с ними, рассматриваемые ***как объекты стандартизации***, относятся в основном к функциональному, оперативному уровню взаимодействия пользователей с системами и программными комплексами.

Стандарты **ISO 15910** и **ISO 18019** являются наиболее современными нормативными документами, регламентирующими процессы ***создания эксплуатационной документации для пользователей сложных программных комплексов***. В них изложены детальные структуры планов документирования, ориентированных на разработчиков документов, соответствующих требованиям на программные комплексы и процедуры контроля реализации плана. Эксплуатационная документация должна проходить ***тестирование и испытания на достоверность***, которые должны быть спланированы и реализованы на базе требований к функциям и характеристикам, а также к применению эксплуатационных процедур реального программного комплекса. Изложены требования и правила оформления ***твердой копии документов*** и правила структурирования и представления схем компонентов, окружения, иллюстрации и состав основного текста документов.

Стандарты представляют разработчикам документации метод определения и применения процесса документирования при создании конкретного программного комплекса. Для соответствия стандартам план производства должен включать спецификацию требований к стилю оформления документов. Стандарты определяют реализацию процесса документирования и могут быть ***адаптированы к условиям и требованиям конкретного программного продукта***. Минимальный состав документации определяется заказчиком (например, с использованием **ISO 12207.2007** или **ISO 6592**), что должно быть учтено при разработке плана документирования.

Описание требований для системы управления и программного продукта, должно содержать действия пользователя, необходимые для работы с системой, ее связи с существующими внешними системами и процедурами и включает ***описания***:

- конкретной эксплуатационной среды и ее характеристики;
- основных компонентов и функций программного продукта, системы и связей между ними;
- внешних интерфейсов программного комплекса и системы;

- возможностей, функций и характеристик программного продукта и системы;
- состава эксплуатационного персонала, его организационной структуры, уровня технической подготовки, обязанностей, взаимодействия;
- форм регистрации обнаруженных дефектов и ошибок;
- соглашений о внесении изменений, возникающих в процессе сопровождения версии программного комплекса;
- концепцию поставки новой или модифицированной версии программного комплекса, эксплуатационный сценарий;
- информацию о взаимодействии пользователей, поставщика, разработчика и предприятия, осуществляющих поддержку программного продукта во время эксплуатационного периода.

В обязанности документаторов входит обеспечение плодотворных ***контактов заказчика с разработчиками*** программного комплекса, гарантирующее, понимание сути и требований к выпускаемой продукции и соответствующих ей аудиторий. Документатор должен предпринять соответствующие шаги по сохранению материалов, представленных заказчиком, обеспечить ***защиту информации о требованиях заказчика***. В ряде случаев требуется сохранить конфиденциальность и секретность предоставленных материалов.

План документирования должен включать ***определение аудитории пользователей документации***, уровня образования, квалификации, способностей, подготовки, опыта пользователей и другие характеристики, связанные с содержанием, структурой и использованием документации. ***План*** должен быть официально согласован и утвержден, подтверждать полный учет и ***тестирование в документах всех требований заказчика к программному комплексу***. План документирования должен быть подготовлен и утвержден до начала разработки документации, чтобы гарантировать согласование всеми сторонами содержания поставленных задач и используемых в проекте методов. После утверждения плана он должен быть доведен до всех заинтересованных сторон, включая разработчиков документации, а также заказчика и субподрядчиков.

Проверка результатов выполнения плана должна проводиться заказчиком с привлечением документаторов. Целью проверки является гарантирование полноты и достоверности представленных материалов и удовлетворения ими возможности выполнения требований заказчика к функциям и характеристикам программного комплекса в

соответствии с условиями договора и плана документирования. Заказчики должны уделять ***особое внимание структуре, полноте и практичности документации*** в соответствии с планом-проспектом ее содержания.

Затраты на создание ***полного комплекта достоверной, эксплуатационной документации*** практически пропорциональны размеру комплекса программ. Эти затраты сопутствуют в некоторой степени всем этапам производства, однако оформление эксплуатационных документов обычно локализуется в специальном этапе работ. Затраты на разработку комплекта ***эксплуатационной документации*** для сложных программных продуктов, подлежащих длительному сопровождению, обычно определяются затратами на объем текста документов ***ориентировочно*** 5 – 10 страниц на тысячу строк программы.

В ***технологической документации,*** обеспечивающей конфигурационное управление и длительное сопровождение версий программного комплекса, необходимы требования, тесты, тексты программ и описания алгоритмов. Это приводит к увеличению объема документации на порядок, т.е. может составлять около 100 страниц документов на тысячу строк программы. Подтверждена наиболее высокая документированность тиражируемых программ реального времени, особенно в тех случаях, когда необходима высокая безопасность и надежность функционирования продукта (до 200 страниц на тысячу строк).

По прецедентам средний объем ***технологической документации*** ~ 50 – 100 страниц на тысячу строк текста программ. При этом затраты на документацию остаются практически пропорциональными размеру комплекса программ. Эта часть документации не подлежит массовому тиражированию и поставке каждому пользователю, что позволяет снижать удельные затраты на ее подготовку. Однако необходимость ее тщательной отработки и высокого соответствия текущей версии программного продукта приводит к большим затратам как при первичном изготовлении технологических документов, так и при их модификации в процессе последующего сопровождения.

Глава 2.8

ИСПЫТАНИЯ, УДОСТОВЕРЕНИЕ КАЧЕСТВА И СЕРТИФИКАЦИЯ СЛОЖНЫХ ЗАКАЗНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Организация и процессы испытаний компонентов и комплексов программ

До начала квалификационного тестирования и испытаний компонентов и комплексов программ подлежат проверке и ***паспортизации средства***, обеспечивающие получение эталонных данных, средства формирования тестов от внешних объектов, средства фиксирования и обработки результатов тестирования. При этом важную роль играет оценка и обеспечение методической достоверности результатов испытаний – рис. 2.16. ***Методическая достоверность испытаний программных комплексов определяется следующими факторами:***

полнотой Программы испытаний и корректностью методик тестирования по охвату возможных сценариев функционирования компонентов и комплекса программ и исходных требований для программного продукта;

достоверностью и точностью эталонных значений требований, с которыми должны сравниваться результаты тестирования испытываемого комплекса программ, которые служат опорными при расчете параметров, зафиксированных в техническом задании и спецификациях требований;

адекватностью и точностью моделей, используемых для формирования сценариев тестов от внешней среды и их реакции на управляющие воздействия программного комплекса;

- точностью и корректностью регистрации и обработки результатов испытаний, а также сравнения полученных данных с требованиями и эталонами технического задания и спецификаций.

Определение задач испытаний предусматривает анализ документов, в которых сформулированы требования к программному продукту, и выделение необходимых работ, чтобы затем проверять функции и характеристики, отражающие эти требования.

Испытания компонентов и сложного комплекса программ включают:

- организацию и процессы испытаний компонентов и комплекса программ:
  - определение задач, процессов и методической достоверности испытаний компонентов и комплекса программ;
  - подготовку к интеграции компонентов, комплекса программ и аппаратуры системы;
- интеграцию и тестирование комплекса программ вне системы:
  - испытания интерфейсов, компонентов и комплекса программ на соответствие требованиям и эталонам;
  - оценка реализуемости и планирование испытаний комплекса программ в составе системы;
  - анализ полноты и корректности документации на комплекс программ;
- приемо-сдаточные испытания программного комплекса в составе системы:
- Программу испытаний комплекса программ на соответствие утвержденным требованиям и эталонам;
- протоколы результатов испытаний программного продукта по разделам Программы испытаний, требований и эталонов;
- комплект адекватной эксплуатационной и технологической документации на программный продукт;
- завершение испытаний на соответствие требованиям и утверждение готовности программного продукта для предъявления заказчику и поставки пользователям;
- Акт о завершения успешных испытаний и выполнения требований заказчика на программный продукт и систему;
- анализ результатов испытаний, усовершенствование методов и процессов тестирования комплексов программ.
- внедрение версии программного продукта для применения в системе и обучение пользователей;
- изменения требований к комплексу программ, создание и сопровождение новых версий программного продукта.

Рис. 2.16.

По списку требований и эталонов может быть определена матрица, составляющая задачи испытаний, которым присваиваются приоритеты. В качестве руководящих принципов для присвоения приоритетов функциям должны использоваться документы, содержащие формулировки требований. В то же время могут возникать и допол-

нительные соображения, оказывающие влияние на выбор приоритетов. Новым функциям и характеристикам имеет смысл посвящать повышенное внимание, нежели слегка модифицированным компонентам. Кроме того, необходимо уделять время на проверку проектных решений и структуры комплекса, а также на тестирование новых компонентов. В дополнение к обычному анализу требований необходимо учесть задачи, которые не имеют прямого отношения к документам с требованиями, но ***влияют на процессы тестирования***:

- составление плана и Программы проведения испытаний;
- разработка тестовых сценариев и динамических моделей внешней среды для генерации тестов;
- отладка тестовых сценариев и генераторов тестов;
- верификация, проверка, выявление и исправление дефектов генераторов тестов;
- определение качества тестов и степени реализации функций и характеристик программного продукта, процесса их сборки и использования;
- пересмотр и корректировка пользовательской документации;
- освоение специалистами по тестированию новых технологий и новых инструментальных средств.

После подготовки такого перечня работ следует его ***ревизовать*** вместе с группой тестирования и провести через полный жизненный цикл, определяя ***дополнительные компоненты и задачи***, которые должны выполняться для корректных испытаний. Такой перечень является динамичным документом, претерпевающим изменения на протяжении всего жизненного цикла тестирования и испытаний по мере роста понимания того, что должно быть сделано для получения программного продукта высокого качества, ***соответствующего требованиям и эталонам заказчика***. Декомпозиция работ часто может выглядеть как иерархический упорядоченный список видов деятельности, в которых каждой задаче присваивается идентификатор. Дескриптор позволяет отслеживать задачу на протяжении всего жизненного цикла разработки и предоставляет возможность связывать измерения трудозатрат или расходов ресурсов с различными задачами. Благодаря этому можно оценивать показатели, характеризующие фактические затраты на протяжении всего проекта.

Факторы, которые следует учитывать перед испытаниями программного комплекса на соответствие требованиям, должны включать:

- идентификатор версии испытываемого программного продукта;
- результаты исправления дефектов, обнаруженных в предыдущей версии, если список дефектов приводится в спецификации, необходимо указать ссылку на требования, функции и характеристики;
- описание внешней среды, используемой при применении программного продукта;
- эксплуатационные документы для пользователей, такие как руководство пользователя, инструкции по инсталляции, особенности версии конкретного продукта.

Функции, характеристики и свойства компонентов и комплекса программ, которые должны тестироваться – это функциональные возможности, отличительные характеристики и риски программного продукта, а также производительность, надежность, безопасность, переносимость, определенные требованиями заказчика. Все, что декларировано заказчиком, должно быть отражено в требованиях, в Программе и плане проведения испытаний с тем, чтобы это протестировать до поставки продукта заказчику. Основным источником, позволяющим фиксировать, что было согласовано с заказчиком, служат ***документы определения требований и эталонов*** (см. главу 1.3). Перечень требований должен сопровождаться технико-экономическим обоснованием и контрольной таблицей для вычисления трудозатрат, необходимых для проведения тестирования и испытаний программного комплекса. Каждая позиция этого перечня должна соответствовать конкретным пунктам документов описания требований, в то же время каждой такой позиции должен соответствовать один или большее число тестов, определенных в ***Программе и спецификациях методик испытаний***.

Внутренние испытания компонентов и программного комплекса (***испытания менеджера проекта***), которые зачастую совмещаются с завершением комплексной отладки, должны оформляться документально и могут являться основанием при предъявлении программного продукта заказчику на квалификационные испытания для завершающего оценивания функций и характеристик качества программного продукта (см. **ISO 12207:2007, ISO 15504, ISO 16326**). Руководитель разработки должен ***оценить уровень реализацию проекта***, состояние комплекса программ, тесты, результаты тестирования и документацию для пользователя, учитывая:

полноту охвата испытаниями всех требований и эталонов к функциональным компонентам и к комплексу в целом;

согласованность с требуемыми заказчиком результатами применения программного продукта;

возможность интеграции и тестирования комплекса программ в составе аппаратуры системы;

возможность функционирования и сопровождения версии программного продукта в соответствии с требованиями контракта.

Этапы тестирования компонентов и комплекса в целом обычно реализуются с позиции последовательного увеличения функциональной сложности тестов и взаимодействия с объектами внешней среды. Этапы и процессы квалификационного тестирования ***с целью формального удостоверения для заказчика достигнутых характеристик и реализации требований*** к комплексу программ и его компонентам в составе системы регламентированы в стандартах **ISO 12207:2007, ISO 15504**. В них выделены ***основные, функциональные*** этапы реализации квалификационного тестирования и испытаний на соответствие требованиям заказчика (см. рис. 2.16).

Квалификационное тестирование функциональных компонентов и комплекса в целом вне системы выполняется для того, чтобы предварительно продемонстрировать заказчику, что реализованы все требования контракта и достигнуто необходимое качество функционирования комплекса программ. Квалификационное тестирование должно ***покрывать все требования*** к функциональным компонентам, требования к комплексу и к интерфейсам. Тестирование функциональных компонентов для каждой конфигурации должно показать, что полностью удовлетворены требования к компонентам, которые должны быть реализованы в данной конфигурации. Ответственными за квалификационное тестирование компонентов не должны быть лица, осуществившие выполнение рабочего проекта или программирование соответствующего компонента. Это не исключает возможность оказания помощи при проведении квалификационного тестирования со стороны лиц, выполнявших рабочий проект или программы, например, путем предоставления тестовых вариантов, основанных на знании ими внутренней реализации функционального компонента или всего комплекса.

Квалификационное тестирование и предварительные испытания должны ***включать работы и объекты***:

- подготовка тестовой версии комплекса программ, содержа-

щей, по крайней мере, два взаимодействующих сложных функциональных компонента, прошедших компонентное тестирование;

- версию комплекса программ, предназначенную для комплексного тестирования, содержащую компоненты, находящиеся под формальным автоматизированным контролем системы управления конфигурацией;
- тестовую версию, собранную из относительно новых функциональных компонентов, которая может быть установлена в тестовой внешней среде таким образом, что она будет стабильно функционировать, получать, поддающиеся интерпретации результаты испытаний;
- заключительную версию, предназначенную для комплексного тестирования и испытаний, содержащую все компоненты, которые войдут в версию, предназначенную для пользователей или установки в системе.

Разработчики должны определить и зарегистрировать процесс подготовки к тестированию, тестовые варианты, сценарии и процедуры, которые должны использоваться для квалификационного тестирования компонента и проследить ***соответствие между тестовыми вариантами и требованиями*** контракта. Они должны подготовить исходные тестовые параметры, необходимые для выполнения тестовых вариантов. Если испытание функционального компонента должно быть засвидетельствовано представителем заказчика, то до его проведения разработчик должен проверить тестовые варианты и тестовые процедуры, чтобы гарантировать, что они полны и точны и что программный комплекс готов для проведения тестирования в присутствии представителя заказчика.

Результаты этих проверок должны быть ***удостоверены в файлах системы управления конфигурацией*** комплекса программ, а тестовые варианты и процедуры соответствующим образом модифицированы для устранения выявленных дефектов. Следует выполнить все необходимые изменения в комплексе программ, предварительно уведомив об этом представителя заказчика, осуществить ***регрессионное тестирование*** в необходимом объеме, модифицировать файлы разработки и другие программные продукты, основываясь на результатах интеграции и тестирования компонентов. Результаты этих действий должны быть включены в отчет для заказчика о результатах проведенных разработчиком предварительных испытаниях комплекса программ. Группа управления проектом должна установить, что вы-

полненное комплексное тестирование достаточно, и дальнейшее продолжение комплексного тестирования, по всей вероятности, не выявит новых дефектов, связанных с интеграцией программных компонентов. Совещание разработчиков и заказчика, принимает решение о том, что критерии завершения квалификационного тестирования вне системы выполнены.

Интеграция и тестирование комплекса программ в составе системы должно содержать их объединение, тестирование полученного в результате комплекса, с целью определения работают ли они корректно вместе, как требуется по контракту. При интеграции ***функциональных компонентов,*** комплексов программ ***реального времени*** целесообразно выделять этапы:

- комплексирование и тестирование функциональных групп программ без взаимодействия с другими компонентами и возможно без подключения к операционной системе реального времени;
- интеграция и тестирование функциональных групп программ с учетом взаимодействия с некоторыми другими компонентами и с базой данных;
- интеграция и тестирование программных компонентов в реальном времени во взаимодействии с другими функциональными компонентами и с компонентами операционной системы и базы данных.

После интегрирования всех откорректированных функциональных компонентов начинаются их тестирование и испытания в составе комплекса программ в целом на соответствие требованиям. Для них наиболее характерны следующие ***этапы интеграции и квалификационного тестирования версии программного продукта в реальном времени***:

- по данным моделирующего стенда или генераторов динамических тестов, имитирующих отдельные объекты и функционирование внешней среды;
- с имитаторами отдельных объектов внешней среды и с реальными воздействиями от операторов-пользователей;
- в полностью адекватной реальной или имитированной внешней среде и с реальными воздействиями от операторов-пользователей на соответствие требованиям.

Квалификационное тестирование при испытаниях системы в целом выполняется, чтобы продемонстрировать заказчику, что ***удовлетворены все требования*** технического задания, – функции и

характеристики качества соответствуют условиям контракта. Оно должно покрывать все требования в спецификациях системы и подсистем, а также требования к интерфейсу с внешней средой. Испытания должны включать тестирование на объектной вычислительной системе или на альтернативной модели системы, одобренной представителем заказчика. В процессе ***системного тестирования*** проверяется интеграция отдельных частей, в совокупности составляющих систему в целом. Тестирование на системном уровне обычно проводится специальной группой специалистов заказчика, которая должна провести анализ с целью определения выполнения требований и компонентов функциональности, которые ***вызывают наибольшее количество проблем***. Анализ результатов тестирования может показать, дало ли выполнение тестовых процедур результат в плане выявления ошибок. ***Тест-менеджер несет ответственность*** за проведение тестирования согласно Программе и плану-графику, а также за распределение и перераспределение работ между тестировщиками для разрешения проблем, возникающих в ходе тестирования. Для эффективного выполнения этой функции тест-менеджеру необходимо динамически отслеживать ход тестирования, готовить и анализировать отчеты.

Разработчики должны участвовать в производстве и регистрации процесса подготовки к тестированию, тестовых вариантов, сценариев и тестовых процедур, которые нужно использовать для полного испытания системы и в прослеживании ***соответствия между тестовыми вариантами и требованиями*** к функциям и характеристикам системы. Каждое проверяемое требование должно соответствовать конкретным, обоснованным характеристикам системы, иметь уникальный для проекта идентификатор, чтобы можно было провести испытание и проследить его выполнение с помощью объективного теста. Для каждого требования должны выбираться квалификационные методы для функциональных подсистем и компонентов программного комплекса, которые необходимо прослеживать в требованиях к системе. Степень детализации требований следует выбирать, учитывая в первую очередь те функции и характеристики качества, которые внесены в условия приемки системы заказчиком, и отдавать ***приоритет*** тем из них, которые заказчик требует обеспечить ***обязательно***.

Все полученные результаты должны быть включены в отчет об испытаниях программного продукта и системы. Если квалификаци-

онное тестирование системы должно быть засвидетельствовано представителем заказчика, то до его проведения разработчик должен проверить тестовые варианты и тестовые процедуры, чтобы гарантировать, что они полны и точны и что система готова для проведения тестирования в присутствии представителя заказчика. Испытания должны быть выполнены в соответствии с ***утвержденными заказчиком*** планом, Программой, тестовыми вариантами, сценариями и процедурами.

База данных должна содержать полный набор данных о дефектах и ошибках, обнаруженных во время тестирования программного комплекса. Сводку дефектов, обнаруженных во время испытаний, целесообразно включать в отчетный доклад. В таком случае всем сторонам, заинтересованным в успехе разработки, не потребуется обращаться в базу данных дефектов за сведениями о ходе работ по испытаниям и о качестве программного продукта. Окончательная версия отчета по результатам анализа дефектов, в котором показаны все обнаруженные дефекты и их окончательные корректировки, а также редакция отчета о не устраненных дефектах и ошибках являются необходимыми дополнениями отчетного доклада о результатах испытаний.

Квалификационное тестирование может завершаться ***приемо-сдаточными испытаниями***, которые подразумевают участие заказчика и возможно пользователей. Приемо-сдаточные испытания, как правило, применяют подмножество тестов, использованных на системном уровне. Обнаруженные в ходе приемо-сдаточных испытаний дефекты документируются в отчетах о проблемах, и определяется их приоритеты. Проблемы, устранение которых невозможно осуществить в отведенные на приемо-сдаточные испытания сроки, следует ***передавать техническому Совету*** для дальнейшего рассмотрения и оценки. По итогам проведения приемо-сдаточных испытаний должен готовиться отчет, который содержит краткое описание произведенных работ и оценку полученных результатов тестирования.

Наиболее полным и разносторонним испытаниям должна подвергаться первая версия программного продукта. При ***испытаниях очередных версий программного продукта*** возможны значительные сокращения объемов тестирования апробированных повторно используемых компонентов. Однако комплексные и завершающие испытания каждой новой версии программного продукта, как правило, должны проводиться в достаточном объеме, гарантирующем провер-

ку выполнения ***всех требований модифицированного технического задания***. Для выявления дефектов в процессе эксплуатации серийных образцов в каждом из них должен быть предусмотрен некоторый минимум средств для оперативной проверки функционирования и обнаружения искажений результатов. Эти средства должны позволять фиксировать условия неправильной работы программ и характер проявления дефектов при применении программного продукта. Последующее исправление ошибок должно проводиться специалистами, осуществляющими сопровождение программного продукта.

Установка приоритетов для тестирования конфигураций версий программных продуктов обычно зависит от ***следующих факторов***:

- частота использования: сколько экземпляров данной конфигурации, скорее всего, будет использоваться;
- риск отказа в работе системы: существуют ли особенно ответственные конфигурации программного продукта для заказчиков с требованиями минимальных рисков;
- оценка вероятности отказа системы: фиксировались ли в прошлом отказы конкретных конфигураций программного продукта и как часто.

Любые ***испытания ограничены допустимым количеством и объемом тестирования***, а также длительностью работы комиссии испытателей, поэтому ***не могут гарантировать абсолютную проверку*** программного продукта на соответствие требованиям к функциям и характеристикам. Для повышения достоверности определения и улучшения оценивания характеристик после внутренних или квалификационных испытаний некоторые экземпляры комплексов программ целесообразно передавать пользователям на ***опытную эксплуатацию в типовых условиях***. Это позволяет более глубоко оценить эксплуатационные характеристики созданного комплекса и оперативно устранять некоторые дефекты и ошибки. Опытную эксплуатацию целесообразно проводить разработчиками с участием испытателей-заказчиков и некоторых пользователей, назначаемых заказчиком. Результаты и характеристики качества опытной эксплуатации после внутренних испытаний могут учитываться при проведении заказчиком квалификационных испытаний для их сокращения.

Для заказчика и пользователей доминирующее значение могут иметь номенклатура и особенности реализации некоторых функций комплекса программ, которые, как правило, требуют наибольших затрат на тестирование и определяют основной эффект от применения

программного продукта, а также потенциальный уровень спроса на рынке. Если затраты на разработку комплекса программ можно оценивать и прогнозировать с некоторой достоверностью, то эффективность применения и особенно будущий спрос на конкретную версию программного продукта со стороны пользователей априори оценить трудно. Такие оценки могут проводиться на основе специальных маркетинговых исследований и опыта эксплуатации аналогичных комплексов программ или достаточно близких их прототипов.

Представленные выше процессы испытаний компонентов и сложных комплексов программ ориентированы на ***наличие конкретного заказчика*** программного продукта и ограниченное число пользователей, контролируемых заказчиком. Несколько иначе организуются испытания коммерческих пакетов прикладных программ, создаваемых по инициативе предприятия или коллектива разработчиков для продажи широкому кругу пользователей при отсутствии конкретного заказчика. Для таких коммерческих комплексов программ принято проводить ***квалификационные испытания на соответствие требованиям***, формализованным руководителем проекта в два последовательных этапа – ***Альфа и Бета тестирование***. Они заключаются в нормальной и форсированной (стрессовой) опытной эксплуатации конечными пользователями оформленного программного продукта в соответствии с эксплуатационной документацией и различаются составом, количеством участвующих пользователей и уровнем их квалификации. При Альфа тестировании привлекаются конечные пользователи, работающие преимущественно в том же предприятии, но не участвовавшие непосредственно в разработке конкретного комплекса программ. Для Бета тестирования привлекаются добровольные пользователи (потенциальные покупатели), которым бесплатно передается версия программного продукта для опытной эксплуатации.

Программа и методики испытаний компонентов и комплексов программ

Оценивание качества и ***соответствия требованиям*** программного продукта при квалификационных и/или приемо-сдаточных испытаниях проводится комиссией заказчика, в которой участвует руководитель (главный менеджер) разработки и некоторые ведущие разработчики, или аттестованной сертификационной лабораторией (см. **ISO 10006**). ***Комиссия при испытаниях должна руководствоваться следующими документами –*** рис. 2.16:

утвержденными заказчиком и согласованными с разработчиком контрактом, техническим заданием и спецификациями требований на программный продукт и систему;

действующими государственными и ведомственными стандартами на жизненный цикл и испытания крупных комплексов программ, на технологическую и эксплуатационную документацию, а также стандартами де-факто, согласованными с заказчиком для использования – профилем стандартов и нормативных документов;

Программой испытаний по всем требованиям контракта, технического задания и спецификаций;

методиками испытаний и матрицей тестов, охватывающими каждый раздел требований технического задания, спецификаций и Программы испытаний;

комплектом адекватной эксплуатационной и технологической документации на программный комплекс.

План испытаний комплекса программ должен описывать порядок квалификационного тестирования функциональных компонентов и подсистем, тестовую внешнюю среду, которая будет использоваться при тестировании, идентифицировать выполняемые тесты и указывать ***план-график*** тестовых действий (см. главу 2.4). Для каждой, предполагаемой тестовой реализации или динамического тестового варианта должны быть указаны:

- используемые версии аппаратных средств;
- интерфейсное оборудование;
- дополнительные внешние устройства;
- генераторы тестовых сценариев или динамических тестовых потоков данных.

Программа испытаний является серией экспериментов и должна разрабатываться с позиции необходимого объема тестирования в процессе проведения испытаний для проверки выполнения всех требований технического задания и соответствия предъявленной документации. Программа испытаний, методики их проведения и оценки результатов, разработанные совместно заказчиком и менеджерами разработки, должны быть согласованы и утверждены. Они должны содержать ***уточнения и детализацию требований*** технического задания и спецификаций для данного программного продукта, а также гарантировать корректную проверку всех заданных функций и характеристик качества. ***Программа испытаний должна содержать следующие четко сформулированные разделы:***

- объект испытаний, его назначение и перечень основных документов, определивших его разработку;
- цель испытаний, с указанием всех требований технического задания, характеристик качества, подлежащих проверке, и ограничений на проведение испытаний программного продукта;
- собственно Программу испытаний, содержащую проверку комплектности спроектированного программного комплекса в соответствие с техническим заданием, план и график проведения тестирования для проверки по всем разделам требований технического задания и дополнительных требований, формализованных отдельными решениями разработчиков и заказчика;
- перечень и содержание методик испытаний, однозначно определяющие все требования, понятия проверяемых функций и характеристик качества, условия и сценарии тестирования, инструментальные средства, используемые для испытаний;
- перечень методик обработки и оценки результатов тестирования программного продукта по каждому разделу Программы испытаний.

Методика испытаний должна содержать: описание организации процесса тестирования, тестовые варианты, сценарии и процедуры, которые используются при испытаниях отдельного функционального компонента или комплекса программ в целом. Каждый тест должен иметь уникальный для данного проекта идентификатор; должны быть представлены инструкции для проведения процедур тестирования; описание аппаратуры и инструментальные средства для реализации тестирования, а также инструкции для динамического и регрессионного тестирования. Кроме того, должны быть приведены ссылки на соответствующие проверяемые требования, указаны условия выполнения (конфигурация аппаратуры и компонентов комплекса программ), входные данные, эталонные и ожидаемые результаты, критерии оценки качества результатов, процедура проведения тестирования для каждого тестового сценария, допущения и ограничения.

Большой объем разнородных данных, получаемых при испытаниях сложных комплексов программ, и разнообразие возможных способов их обработки, интерпретации и оценки приводят к тому, что важнейшими факторами становятся ***методики обработки и оценки результатов, а также протоколы проверки по пунктам Программы испытаний.*** В соответствии с методиками испытаний средства автоматизации должны обеспечивать всю полноту проверок характеристик по каждому разделу методик. ***Результаты испытаний фик-***

сируются в протоколах (см. **ISO 12119)** , которые обычно содержат следующие разделы:

- назначение тестирования и раздел требований технического задания, по которому проводились испытания;
- указания разделов методик в соответствии, с которыми проводились испытания, обработка и оценка результатов;
- условия и сценарии проведения тестирования и характеристики исходных данных при динамическом тестировании;
- обобщенные результаты испытаний с оценкой их на соответствие требованиям технического задания и другим руководящим документам, а также технической и эксплуатационной документации;
- описание отличий тестовой и реальной эксплуатационной среды;
- описание обнаруженных дефектов и ошибок и рекомендуемых улучшений в испытываемом комплексе программ;
- выводы о результатах испытаний и о соответствии созданного программного комплекса или его функционального компонента определенному разделу требований технического задания и исходных спецификаций.

Протоколы по всей программе испытаний ***обобщаются в акте,*** в результате чего делается ***заключение о соответствии системы требованиям*** заказчика и о завершении работы с положительным или отрицательным итогом. При выполнении всех требований технического задания заказчик обязан принять программный продукт, и проект считается завершенным.

Завершение испытаний сложных заказных программных продуктов

Завершение испытаний программного комплекса в составе системы должно удостоверить и зафиксировать следующие ***процессы и результаты***:

- завершена разработка всех функций комплекса программ и исправление всех выявленных ошибок;
- все функциональные компоненты размещены под формальный автоматизированный контроль системы управления конфигурацией компонентов проекта;
- завершено компонентное тестирование всех функций и исправление всех выявленных дефектов версии программного продукта;
- подготовлена полная версия программного комплекса с кон-

тролируемыми изменениями после испытаний;

- версия программного комплекса, переданная на испытания, сопровождается технологической и эксплуатационной документацией, перечнем изменений, содержит список отчетов о дефектах, которые исправлены в версии;
- предоставлена контролируемая полная версия программного комплекса, которая установлена в тестовой среде, стабильно функционирует и позволяет получать поддающиеся интерпретации результаты испытаний;
- проведены совещания менеджеров и специалистов, посвященные управлению незавершенными работами по устранению дефектов и оценки времени для исправления дефектов;
- в предшествующие несколько недель не произошло ни одного сбоя, остановки, неожиданного прекращения процесса или другой аномалии функционирования комплекса программ на объектном компьютере в системе;
- анализ динамического функционирования показал, что комплекс программ достиг приемлемого уровня качества, стабильности, надежности и безопасности;
- оценки покрытия требований допустимых рисков показали, что все риски сокращены до допустимых пределов;
- менеджеры и группа управления системой установила, что программный продукт, как это определено в ходе заключительного цикла испытаний, удовлетворяет требованиям заказчика и пользователей;
- проведено совещание менеджеров производства с заказчиком и установлено, что ***критерии завершения испытаний программного продукта выполнены***.

Завершаются испытания предъявлением заказчику на утверждение ***комплекта документов***, содержащих ***результаты комплексных испытаний*** версии программного продукта:

- откорректированные тексты программ и данных на языке программирования и в объектном коде, а также полные спецификации требований на программные компоненты и комплекс в целом после полного завершения тестирования и испытаний;
- Программу испытаний программного продукта по всем требованиям технического задания;
- комплект методик испытаний и обработки результатов по всем разделам Программы испытаний;

- тесты, сценарии и генераторы тестовых данных, использованные для испытаний программных компонентов и версии продукта в целом;
- результаты и протоколы квалификационного тестирования, функциональные и конструктивные характеристики программного продукта в реальной внешней среде;
- отчет о подтверждении заданного качества, полные характеристики достигнутого качества функционирования, а также степени покрытия тестами спецификации требований к программному продукту;
- план, методики и средства автоматизации обучения заказчика и пользователей применению испытанной версии программного продукта;
- комплект эксплуатационной документации, описание программного продукта и руководство пользователя в соответствии с условиями контракта;
- технические условия на версию программного продукта, базу данных управления конфигурацией и эксплуатационную документацию для тиражирования и серийного производства;
- руководство по генерации и инсталляции пользовательских версий и загрузке базы данных в соответствии с условиями и характеристиками внешней среды;
- отчет о технико-экономических показателях завершенного проекта версии программного продукта, выполнении планов и использованных ресурсах;
- ***акт о завершении испытаний и готовности к поставке*** и/или предъявлению для сертификационных испытаний версии программного продукта.

Чтобы исключить напрасную трату времени на более поздних этапах, целесообразно подготовить шаблоны этих документов на этапе составления плана проведения испытаний. Определение и согласование с заказчиком на раннем этапе жизненного цикла состава документов при завершении испытаний, может упростить построение плана работ. На завершающих этапах испытаний, когда, как правило, на группу тестирования оказывают давление руководители для ускорения с окончанием работ, разработчикам полезно иметь в своем распоряжении ***документ***, в котором оговариваются все требования и дополнения заказчика, которые планировалось испытать. Специалисты склонны забывать, какие соглашения и требования были достигнуты,

а также планы проведения испытаний. Если проблема возникнет в области, которая не подвергалась испытаниям в соответствии планом и Программой, она должна быть устранена и проверена после выпуска очередной версии программного продукта. Поскольку стоимость исправления дефектов после сдачи программного продукта в эксплуатации велика, важно определять по возможности точную оценку рисков, прежде чем принимать решение проводить дополнительных испытаний того или иного свойства или конфигурации версии программного продукта.

Отчетный доклад о результатах испытаний должен содержать перечень ***всех не устраненных дефектов*** с соглашением и планом того, будут ли они исправлены в более поздних версиях или же их исправление откладывается на неопределенное время. Дальнейшие версии программного продукта могут содержать не устраненные дефекты до тех пор, пока они не будут исправлены. Необходимо вести наблюдение за такими дефектами, чтобы не тратить дополнительные усилия на их выявление в будущих версиях продукта. В плане проведения испытаний будущих версий должны присутствовать ссылки на список не устраненных дефектов. Это позволит тестировщикам учитывать, что их ожидает во время испытаний новой версии программного продукта.

Критерий выхода из испытаний и утверждения готовности версии заказного программного продукта для поставки, которые могут использоваться для принятия решения о прекращении испытаний:

- завершены все запланированные циклы тестирования и план проведения испытаний выполнен, тестовое покрытие лучше, чем в случае, когда просто не хватило времени для доведения тестирования до логического конца;
- профиль устраненных дефектов соответствует критерию выхода из испытаний и возможное количество не устраненных ошибок на тысячу строк программного кода достигнуто;
- время, отведенное на тестирование и испытания, истекло, хотя неизвестно достигнутое качество программного продукта.

Принимая решение прекратить работы по испытаниям, следует учитывать несколько факторов, которые играют важную роль при ***оценке неготовности заказного программного продукта*** к поставкам:

- количество дефектов, обнаруженных в процессе тестирова-

ния, которые остаются неисправленными;

- общее количество возможных предсказуемых дефектов, которые не были исправлены;
- процентное отношение количества тестов, которые завершились успешным исходом и устранением дефектов к числу всех запланированных тестов;
- количество тестов, реализация которых не может быть выполнена, поскольку они заблокированы дефектами.

С целью выработки ***оценки готовности версии*** отлаженного программного продукта следует проводить специальные ***совещания менеджеров - разработчиков и заказчика***. В некоторых предприятиях оценку готовности определяет группа тестирования; в других организациях совещание проводит руководитель проекта системы либо руководитель разработки и заказчик программного продукта. В любом случае в задачу группы тестирования и испытаний входит представление результатов и выработка ***рекомендаций относительно готовности продукта к поставкам***. Хотя оценка готовности может проводиться формально, в любом случае следует фиксировать имена участников и принятое ими ***решение, касающееся поставок заказного программного продукта***. Чтобы оценка готовности содержала результаты и рекомендации предприятия, проводившего испытания, эта оценка должна содержать ссылку на отчетный доклад по результатам испытаний.

Критерии успешного или неудачного прохождения испытаний и отдельных тестов для программных компонентов должны быть включены в тестовые сценарии, в связи с чем их следует помещать в план проведения испытаний. С самого начала проекта необходимо определить, что следует считать успешным прохождением стадии испытаний и ***соответствия исходным требованиям и эталонам*** заказчика, когда можно прекратить испытания продукта и начать его поставку. В некоторых случаях критерий выхода из испытаний определяется в Программе испытаний или в документе, который содержит формулировки требований, предъявляемых к программному продукту. Вне зависимости от того, кто служит источником этих критериев, следует давать их четкую формулировку в плане и Программе проведения испытаний.

После завершения испытаний новой версий программного продукта обычно осуществляется процесс ее внедрение для применения. Это производится, как правило, в два этапа; силами разра-

ботчиков версии в целях обкатки, проверки и выявления ошибок в изменениях на этапе опытной эксплуатации, и посредством использования специализированных коллективов сопровождения для тиражирования и распространения версий программного продукта. Основные обязанности специалистов сводятся к передаче физических носителей с текстами программ и комплектом эксплуатационной документации, а также к проведению консультаций для выделенной группы специалистов заказчика и пользователей. Разработчики и испытатели в этом случае получают возможность непосредственно ***контролировать работу пользователей*** с системой и документацией, что обеспечивает высокую оперативность отработки замечаний и рекламаций, формирование квалифицированных предложений для изменений, оценку эффективности применения версии программного продукта. Кроме того, должны разрабатываться учебно-методический план, подготавливаться учебные пособия, необходимые для обучения пользователей, а также проводиться обучение выделенной группы специалистов, ответственных за последующее обучение коллективов пользователей.

По окончании производства версии программного продукта специалистам полезно ***изучить ход выполнения Программы испытаний***, чтобы определить, каким образом можно ее усовершенствовать и в следующем проекте получить преимущества. Группа тестирования должна освоить непрерывный итерационный процесс ***изучения полученных уроков как часть развития культуры тестирования***.

После передачи версии программного продукта в эксплуатацию затраты ресурсов на обнаружение и первичную квалификацию дефектов несут в основном непосредственные пользователи. На разработчиков (поставщиков) программного продукта возлагаются затраты на анализ и локализацию источников и причин дефектов и их устранение. Эти затраты зависят от характеристик выявляемых дефектов, от масштаба комплекса программ, организации и технологии его разработки, инструментальной оснащенности сопровождения, квалификации специалистов, а также от тиража и активности применения данного программного продукта. Гибкость и кажущаяся простота изменения программ значительно затрудняют оценки необходимого, иногда весьма значительного, финансирования на тестирование и испытания, а также на сопровождение и проведение жесткого конфигурационного управления версиями программного продукта.

При сопоставлении результатов оценивания функций и характеристик качества с требованиями технического задания и спецификаций разработчик или поставщик ***обязан удовлетворять требования заказчика только в пределах согласованных параметров*** модели внешней среды и системы. Оценивание функционирования комплекса программ за этими пределам должно дополнительно согласовываться испытателями с заказчиком и производителями. При этом не выполнение требований может квалифицироваться как их расширение за пределы, ограниченные контрактом, и не учитываться при оценивании заказчиком характеристик качества программного продукта, или как дополнительные работы, подлежащие соответствующему финансированию со стороны заказчика для доработки комплекса программ с целью удовлетворения этих требований.

Организация сертификации сложных заказных программных продуктов

Потребителей – заказчиков программных продуктов управляющих систем интересует, прежде всего, ***качество готового продукта*** и обычно не очень беспокоит, как оно достигнуто. Однако это качество должно быть ответственно ***удостоверено и гарантировано*** компетентными, независимыми организациями. Гарантирование качества продукции возможно посредством сертификационных испытаний ***процессов производства*** комплексов программ и/или испытаний их результатов – ***готовых программных продуктов*** [22]. Рассматриваемые далее заказные программные продукты для систем управления и обработки информации в реальном времени активно применяются в сложных критических и ответственных системах динамического управления объектами. Проектирование и производство таких программных продуктов, обычно требуется заказчиками базировать на международных стандартах, охватывающих весь их жизненный цикл.

Сертификация – это процедуры подтверждения соответствия продукции требованиям и стандартам, установленным заказчиком, независимым от изготовителя и потребителя. Сертификационные испытания должны ***технически и юридически удостоверять*** в письменной форме, что состояние продукции, процессов его производства и системы менеджмента качества способны обеспечить требуемое качество и стабильность характеристик изготовляемой продукции любыми ***двумя методами*** (рис.2.17).

Рис.2.17

Первый метод должен обеспечивать высокое качество выполнения всего технологического процесса проектирования и производства, и тем самым минимум экономических потерь от брака, что особенно важно при создании сложных дорогих систем. Результаты испытаний качества ***процессов проектирования и производства*** трудно измерять количественными критериями, и обычно характеризуются рядом требований к качественному выполнению наборов стандартизированных производственных процессов. Они оцениваются свойствами различных процессов, непосредственно отражающимися на характеристиках качества программного продукта, однако при этом нет гарантии адекватного и однозначного требуемого качества конечного продукта. При производстве этот метод может приводить к неконтролируемому, неизвестному качеству компонентов и комплексов программ в целом, и к значительным экономическим потерям за счет затрат на создание не пригодного к использованию продукта (брака), что может быть дорого для сложных систем.

Второй метод сертификации акцентирован на анализе, контроле и удостоверении качества готового программного продукта, которое удостоверяется при его испытаниях. Отсутствие или недостатки системы обеспечения качества в технологическом процессе разработки могут приводить к длительному итерационному процессу доработок и повторных испытаний. При сертификационных испытаниях ***готового программного продукта*** могут использоваться его стандар-

тизированные количественные и качественные критерии качества и характеристики, которые непосредственно отражают функции и свойства продукции, интересующие заказчика и потребителей, их можно измерить и достоверно установить реальные значения.

Соответственно можно выделить ***два вида сертификационных испытаний***: ***технологий*** обеспечения жизненного цикла программных комплексов, поддержанных регламентированными системами качества; и испытаний готового программного ***продукта*** с полным комплектом эксплуатационной документации (см. рис. 2.17). Взаимосвязь качества разработанных комплексов программ с качеством технологии их создания и с затратами на производство становится особенно существенной при необходимости получения ***критического заказного программного продукта реального времени*** с особенно высокими значениями характеристик качества при ограниченных ресурсах [35]. Этот вид комплексной сертификации должен обеспечивать контроль реализации требований алгоритмической и функциональной корректности программного продукта, что особенно важно в программных комплексах для ***обеспечения функциональной безопасности применения сложных управляющих систем***.

Сертификация производства продукции различных видов регламентирована стандартами: **ГОСТ Р ИСО 9001:2001; ГОСТ Р 40.003: 2005** и **ГОСТ Р ИСО 19011: 2003,** а также ***комплексом международных стандартов*** создания и жизненного цикла программных продуктов и их компонентов (см. Приложение 1). Они акцентированы на ***Системе менеджмента качества производства.*** При этом ***сертификация производства*** определена как ***процедура подтверждения соответствия***, посредством которой независимая от изготовителя (продавца, исполнителя) и потребителя (покупателя) организация удостоверяет в письменной форме, что состояние производства (системы менеджмента качества производства) способно обеспечить требуемое качество и стабильность характеристик изготовляемой конкретной продукции.

Испытания производства и программных продуктов должны осуществлять эксперты (аудиторы) по сертификации производств, зарегистрированные в Регистре системы сертификации персонала – ***сертифицированные специалисты***. Область сертификации определяет заказчик по согласованию с председателем комиссии органа по сертификации конкретной продукции. На практике акцент, распределение ресурсов и усилий на два вида сертификации зависят от осо-

бенностей характеристик комплекса программ, квалификации коллектива специалистов – разработчиков, требований заказчика – потребителей и наличия у сертификационной лаборатории соответствующей тематической квалификации. Для этого организация и процессы сертификации должны специализироваться на определенные классы программных продуктов, предусматривать соответствующие технологические работы и документы, обеспечивающие создание продукта требуемого качества [22].

В двух последующих разделах приводятся ***примеры*** процедур и содержания документов для сертификации технологических процессов производства и сертификации качества готовых программных продуктов. Эти разделы могут использоваться в качестве методической базы для подготовки руководящих документов и инструкций при производстве конкретных программных продуктов и их сертификации, а также для подготовки специалистов – сертификаторов.

Сертификация технологических процессов производства сложных заказных программных продуктов

Для сертификации технологических процессов предприятие должно установить ***перечень процессов и документов***, которые необходимы для управления производством программных продуктов, а также для обеспечения уверенности в соответствии продукции требованиям заказчика и стандартов. Головным стандартом регламентирующим производство может быть стандарт на ***основе модели*** – **CMMI** (см.главу 1.1). Достижение высоких значений качества комплексов программ существенно зависит от качества – ***зрелости*** технологии и инструментальных средств, используемых разработчиками при проектировании и производстве программного продукта. Оценивание уровня зрелости технологической базы жизненного цикла (ЖЦ) позволяет прогнозировать возможное качество продукта и ориентировать заказчика и пользователей при выборе разработчика и поставщика проекта с требуемыми характеристиками. Поэтому определение уровня ***зрелости*** технологической поддержки процессов жизненного цикла, организационного и инструментального обеспечения, непосредственно ***связано с оцениванием реальных или возможных характеристик качества производства*** конкретного программного продукта.

Практически все требования к производству программного продукта в ***модели*** **CMMI** ***соответствуют*** регламентированным и детализированным требованиям в стандартах **ISO 9001:2000, ISO**

12207:2007 и базовых компонентах ***стандартов жизненного цикла сложных*** комплексов программ. На их основе формируются процедуры сертификации производства программных продуктов. Для сертификации производства и системы менеджмента качества предприятия, необходима ***четкая организация документирования производства***. Входные документы для производства должны включать все требования, существенные для проектирования и производства программного продукта. Выходные данные процесса проектирования и/или производства должны быть зарегистрированы в документах в форме, дающей возможность проверки их по отношению к входным требованиям. Документы, содержащие выходные данные, должны быть утверждены до их применения при сертификации.

Документ, содержащий ***результаты сертификационных испытаний*** производства и системы менеджмента качества программных продуктов, должен включать Программу и методики сертификационных испытаний – аудита производства предприятия, протоколы и отчет аудиторов о результатах испытаний качества производства программного продукта (см. рис. 2.16). В документе должны быть представлены результаты аудита, выводы и рекомендации комиссии, оформленные в виде акта удостоверяющего качество производства. Завершение сертификации, выдача и регистрация сертификата по результатам аудита – испытаний производства программного продукта, должен определять достаточность качества программного продукта для поставки потребителям и обеспечения его жизненного цикла, а также для регистрации лицензии на применение знаков соответствия.

Сертификация проектирования и производства программных продуктов высокого качества включает следующие основные процессы:

2.1. Определение конкретной среды, процессов производства и основных характеристик программного продукта (рис. 2.18).

2.2. Подготовка к сертификации производства и системы качества программных комплексов и предприятия.

2.3. Подготовка и документирование организации процессов сертификации производства и системы менеджмента качества предприятия.

2.4. Организация сертификационных испытаний и системы менеджмента управления качеством производства программных продуктов.

2.5. Анализ результатов и завершение сертификационных испы-

таний производства программного продукта (рис. 2.19).

Требования к составу документов для организации сертифицируемого производства включают:

- комплект должностных инструкций, определяющих ответственность, и порядок взаимодействия специалистов, для производства программного продукта;
  - положение о подразделениях и должностные инструкции, обязанности и полномочия специалистов, реализующих производство программных продуктов;
  - набор характеристик комплекса программ для сертификационных испытаний;
- состав документации процессов и результатов сертификации – аудита технологии и системы качества производства программных продуктов:
  - заявка клиента на проведение сертификации производства программного продукта;
  - задание клиента на проведение сертификационных испытаний производства программного продукта;
  - план сертификации производства программного продукта;
- документы сертификационной лаборатории:
  - требования к Программе сертификационных испытаний производства программного продукта;
  - Программа испытаний и методики аудита – испытаний процессов производства программного продукта;
  - требования к документации результатов испытаний производства программного продукта;
  - состав технологических средства автоматизации и Программа сертификационных испытаний производства;
  - методики испытаний по каждому разделу требований процессов производства программного продукта;
  - отчеты выполнения и результаты испытаний производства;
  - заключение по результатам сертификационных испытаний процессов производства программного продукта;
  - отчет клиенту – заявителю о проверках организации сертификационных испытаний – аудита и системы качества производства программного продукта.

Рис. 2.18.

Результаты сертификационных испытаний проектирования, производства и системы менеджмента качества программных продуктов включают:

- Программу и методики сертификационных испытаний – аудита производства и системы менеджмента качества предприятия программного продукта:
  - результаты проведение аудита – испытаний на месте производства и подготовка акта аудита предприятия и контроля качества производства программного продукта;
  - протоколы и отчет главного аудитора о результатах испытаний качества производства программного продукта;
  - результаты оценивания достаточности испытаний качества и возможности завершить процесс тестирования производства;
- результаты аудита, выводы и рекомендации комиссии, оформленные в виде акта удовлетворительного качества производства программного продукта:
  - завершение сертификации, выдача и регистрация сертификата по результатам аудита – испытаний процессов производства программного продукта;
  - сертификат производства программного продукта, достаточный для поставки потребителям и обеспечения его жизненного цикла, лицензии на применение знаков соответствия;
  - результаты эпизодического инспекционного контроля состояния сертифицированного производства программного продукта.

Рис. 2.19

Сертификация качества готовых сложных заказных программных продуктов

При сертификации заказных программных продуктов и разработке тестов сертификаторы должны иметь четкое ***представление о потребностях заказчика***. Необходимо изучить системные требования, сценарии использования системы и/или описание назначения системы для того, чтобы лучше понять цель разработки программного продукта, выбор методов и средств его тестирования.

Требования к тестам и документы должны содержать подробный перечень того, что и как должно быть испытано [16, 13, 35]. ***Стратегия сертификационных испытаний*** – это документ, отражающий совокупность выбранных методов, требований и решений, вытекающих из целей и задач проекта и его тестирования, общие

правила и принципы, способствующие достижению целей разработки программного продукта высокого качества.

При сертификационных испытаниях программного продукта целесообразно выборочно или полностью использовать результаты предварительных испытаний с учетом их полноты и достоверности. Завершаются предварительные испытания разработчиков предъявлением заказчику на утверждение комплекта документов, содержащих результаты, необходимые для сертификационных испытаний программного продукта. Результатом сертификационных испытаний должен быть комплект отчетных документов и подтверждение результатов, зафиксированных при предварительных испытаниях.

Утверждение этого комплекта документов для конкретного программного продукта дает право на присвоение ему ***сертификата и знака качества***. Отчетный доклад о результатах испытаний должен содержать перечень ***всех не устраненных дефектов***, с соглашением и планом того, будут ли они исправлены в более поздних версиях или же их исправление откладывается на неопределенное время.

Снятие с эксплуатации и развития версий сертифицированного программного продукта должно быть подготовлены анализом, обосновывающим это решение. При снятии программного продукта с сопровождения следует определить необходимые для этого действия, а затем разработать и документально оформить этапы работ, обеспечивающие их выполнение. Должны быть предусмотрены возможности ***доступа к полным архивным документам*** снятого с сопровождения программного продукта. ***Сертификация готовых заказных программных продуктов высокого качества включает следующие основные процессы:***

3.1. Формирование требований к характеристикам качества для сертификации программного продукта (рис. 2.20).

3.2. Оценка процессов и ограничений сертификационных испытаний программных продуктов на соответствие требованиям.

3.3. Организация и планирование сертификационных испытаний программных продуктов и компонентов на соответствие требованиям.

3.4. Стратегии испытаний качества программных продуктов.

3.5. Подготовка тестов для сертификационных испытаний программных продуктов.

3.6. Предварительные испытания и опытная эксплуатация разработчиками качества программного продукта.

3.7. Завершение сертификационных испытаний и удостоверение качества программных продуктов (рис. 2.21).

Организация сертификационных испытаний компонентов и программного продукта на соответствие требованиям включают:

- **требования** к видам и тестам испытаний компонентов и программных продуктов:
  - комплект требований к адекватной эксплуатационной и технологической документации на компоненты и программный продукт;
  - оценка основных лимитирующих ресурсов испытаний качества – трудозатрат специалистов, и ограничений на сроки производства программного продукта;
- **план** сертификационных испытаний компонентов и программного продукта:
  - график проведения тестовых процедур при испытаниях компонентов и программного продукта;
  - верификация требований к тестам к компонентам и комплексу программ;
  - требования необходимого покрытия тестами требований к компонентам и программному продукту;
  - распределение ресурсов для испытаний в зависимости от характеристик выявляемых дефектов и реализуемых изменений в программах и данных;
  - оценка необходимых объемов и сроков для завершения сертификационных испытаний программного продукта.

Рис.2.20.

Современные заказные программные продукты для управления и обработки информации активно применяются в сложных критических и ответственных системах динамического управления объектами различного назначения. Ущерб от дефектов программных продуктов таких систем может определяться огромной их стоимостью и жизней пользователей. На реализацию требуемого качества требуются ресурсы, которые могут быть соизмеримыми с первичными затратами на весь процесс проектирования и производства системы и программного продукта, а также на подготовку и воспитание профессиональных квалифицированных коллективов специалистов в области индустрии сложных программных продуктов высокого качества.

Результаты сертификационных испытаний сложного заказного программного продукта включают:

- исходные документы заявителя для выполнения сертификации заказного программного продукта:
  - техническое задание – требования к функциям, характеристикам, качеству программного продукта, системы и внешней среды;
  - договор заказчика с производителем на качество программного продукта;
  - полные тексты программ, содержание базы данных и технологической документации программного продукта;
  - результаты испытания эксплуатационной документации на соответствие требованиям к программному продукту;
  - комплект эксплуатационных документов, поставляемых заказчику и пользователям для применения программного продукта;
  - план, Программа и методики испытаний, применения и оценки качества программного продукта;
  - руководство по генерации и инсталляции пользовательских версий и загрузке базы данных в соответствии с условиями и характеристиками внешней среды;
- договор заявителя с сертифицирующей организацией на проведение испытаний версии заказного программного продукта:
  - отчет сертификаторов о реализации Программы и методик проведения сертификационных испытаний программного продукта в соответствии с Договором на сертификацию с заявителем;
  - результаты аттестации имитаторов внешней среды и генераторов динамических тестов для сертификационных испытаний программного продукта;
  - результаты выполнения планов и методик сертификационных испытаний, протоколы испытаний предъявляемым требованиям, утвержденным сертификаторами и согласованным с заявителями;
  - протоколы достигнутых при сертификационных испытаниях характеристик качества программного продукта;
- **Акт** результатов сертификационных испытаний заказного программного продукта, реальные характеристики программного продукта, выводы о их соответствии требованиям к характеристикам заказчика программного продукта;
  - **сертификат** заказного программного продукта, лицензия на применение знаков соответствия;
  - **удостоверение** для поставки и применения пользователями сертифицированного программного продукта.

Рис. 2.21.

Для достижения и удостоверения требуемого высокого качества необходимо создание и ***внедрение методов и средств автоматизации сертификации производственных процессов*** и их результатов – ***программных продуктов*** сложных заказных систем.

ПРИЛОЖЕНИЕ

МЕЖДУНАРОДНЫЕ И ГОСУДАРСТВЕННЫЕ СТАНДАРТЫ, РЕГЛАМЕНТИРУЮЩИЕ ПРОЕКТИРОВАНИЕ И ПРОИЗВОДСТВО СЛОЖНЫХ ЗАКАЗНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

1. **ГОСТ Р ИСО/МЭК 12207.2007** – Системная и программная инженерия, процессы жизненного цикла систем и программных комплексов; **ISO 12207:2007.**
2. **CMMI – Capability Maturity Model Integration for Product and Process Development –** Интегрированная модель оценивания зрелости продуктов и процессов разработки программных средств.
3. **ISO 19759:2005. SWEBOK.** Свод знаний о программной инженерии.
4. **ISO 15288:2002.** Системная инженерия. Процессы жизненного цикла систем.
5. **ISO 19760:2003.** – Системная инженерия. Руководство по применению стандарта **ISO 15288**.
6. **ISO 12207:1995.** (ГОСТ Р – 1999). ИТ. Процессы жизненного цикла программных средств.
7. **ISO 15271:1998.** (ГОСТ Р – 2002). ИТ. Руководство по применению ISO 12207.
8. **ISO 16326:1999.** (ГОСТ Р – 2002). ИТ. Руководство по применению ISO 12207 при административном управлении проектами.
9. **ISO 15504 – 1-5: 2003-2006.** ИТ. Аттестация процессов. Ч.1. Концепция и словарь. Ч.2. Подготовка к аттестации. Ч.3. Руководство по проведению аттестации. Ч.4. Руководство для пользователей по усовершенствованию процессов и определению зрелости процессов. Ч.5. Образец модели аттестации процессов.
10. **ГОСТ Р 51904 – 2002**. Программное обеспечение встроенных систем. Общие требования к разработке и документированию.
11. **ISO 9000:2000.** (ГОСТ Р – 2001). Система менеджмента (административного управления) качества. Основы и словарь.
12. **ISO 9001:2000.** (ГОСТ Р – 2001). Система менеджмента (административного управления) качества. Требования.

13. **ISO 9004:2000.** (ГОСТ Р – 2001). Система менеджмента (административного управления) качества. Руководство по улучшению деятельности.
14. **ISO 90003:2004** – Руководство по организации применения стандарта **ISO 9001:2000** для программных средств.
15. **ISO 10007: 1995** - Административное управление качеством. Руководящие указания при управлении конфигурацией.
16. **ISO 12182:1998.** (ГОСТ Р– 2002). ИТ. Классификация программных средств.
17. **ISO 9126:1991.** (ГОСТ – 1993). ИТ. Оценка программного продукта. Характеристики качества и руководство по их применению.
18. **ISO 14598-1-6:1998-2000.** Оценивание программного продукта. Ч.1. Общий обзор. Ч. 2. Планирование и управление. Ч. 3. Процессы для разработчиков. Ч.4. Процессы для покупателей. Ч.5. Процессы для оценщиков. Ч. 6. Документирование и оценивание модулей.
19. **ISO 9126-1-4: 2002**. ИТ. ТО. Качество программных средств: Ч.1. Модель качества. Ч.2. Внешние метрики. Ч. 3. Внутренние метрики. Ч. 4. Метрики качества в использовании.
20. **ISO 25000:2005** ТО. – Руководство для применения новой серии стандартов по качеству программных средств на базе обобщения стандартов ISO 9126:1-4: 2002 и ISO 14598:1-6:1998-2000.
21. **ISO 15939: 2002** – Процесс измерения программных средств.
22. **IEC 61508:1-6: 1998-2000.** Функциональная безопасность электрических / электронных и программируемых электронных систем. Часть 3. Требования к программному обеспечению. Часть 6. Руководство по применению стандартов IEC 61508-2 и IEC 61508-3.
23. **ISO 15408 -1-3. 1999.** (ГОСТ Р – 2002). Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Ч.1. Введение и общая модель. Ч. 2. Требования к функциональной безопасности. Ч.3. Требования к доверию безопасности.
24. **ISO 13335 - 1-5. 1996-1998.** ИТ. ТО. Руководство по управлению безопасностью. Ч.1. Концепция и модели обеспечения безопасности информационных технологий. Ч.2. Планирование и управление безопасностью информационных технологий. Ч.3. Техника

управления безопасностью ИТ. Ч.4. Селекция (выбор) средств обеспечения безопасности. Ч.5. Безопасность внешних связей.
25. **ISO 10181: 1-7. BOC. 1996-1998.** Структура работ по безопасности в открытых системах. Ч.1. Обзор. Ч.2. Структура работ по аутентификации. Ч.3. Структура работ по управлению доступом. Ч.4. Структура работ по безотказности. Ч.5. Структура работ по конфиденциальности. Ч.6. Структура работ по обеспечению целостности. Ч.7. Структура работ по проведению аудита на безопасность.
26. **ISO 16085:2005.** SE. Процессы жизненного цикла программных средств. Управление рисками.
27. **ISO 14252:1996** (IEEE 1003.0). Руководство по функциональной среде открытых систем POSIX.
28. **ISO 9945-1:1990** (IEEE 1003.1). ИТ. Интерфейсы переносимых операционных систем. Ч.1. Интерфейсы систем прикладных программ (язык Си).
29. **ISO 9945-2:1992** (IEEE 1003.2). ИТ. Интерфейсы переносимых операционных систем. Часть 2. Команды управления и сервисные программы.
30. **ISO 13210:1994.** ИТ. Методы тестирования для измерения соответствия стандартам POSIX.
31. **ISO 14756: 1999.** ИТ. Измерение и оценивание производительности программных средств компьютерных вычислительных систем.
32. **ISO 12119:1994.** (ГОСТ Р – 2000 г). ИТ. Требования к качеству и тестирование.
33. **ANSI/IEEE 829 - 1983**. Документация при тестировании программ.
34. **ANSI/IEEE 1008 - 1986**. Тестирование программных модулей и компонентов программных средств.
35. **ANSI/IEEE 1012 - 1986**. Планирование верификации и подтверждения достоверности качества (валидации) программных средств.
36. **ISO 14764: 1999.** (ГОСТ Р – 2002). ИТ. Сопровождение программных средств.
37. **ISO 15846:1998.** ТО. Процессы жизненного цикла программных средств. Конфигурационное управление программными средствами.

38. **ISO 15910:1999.** (ГОСТ Р – 2002) ИТ. Пользовательская документация программных средств.
39. **ISO 18019:2004** ИТ. Руководство по разработке пользовательской документации на прикладные программные средства для офисов, бизнеса и профессиональных применений.
40. **ISO 6592:2000.** ОИ. Руководство по документации для вычислительных систем.
41. **ISO 9294:1990.** (ГОСТ–1993 г). ТО. ИТ. Руководство по управлению документированием программного обеспечения.
42. **ГОСТ Р 51901-2002.** Управление надежностью. Анализ рисков технологических систем.
43. **ISO 16085: 2004** – Характеристики процессов управление рисками при разработке, применении и сопровождении программных средств.
44. **ISO 14143: 1-5: 1998 – 2004.** ИТ. Измерение программных средств. Измерение функционального размера. Ч.1. 1998. Определение концепции. Ч.2. 2002. Оценивание соответствия методов измерения размера программных средств стандарту ISO 14143:1:1998. Ч.3. 2003. Верификация методов измерения функционального размера. Ч.4.2002. Эталонная модель. Ч.5. 2004. Определение функциональных доменов для использования при измерении функционального размера.
45. **ISO 20926:2003.** Руководство по практическому методу измерения функционального размера программных средств.
46. **ISO 20968:2002.** Руководство по расчетам на основе анализа функциональных точек – Марк II.

ЛИТЕРАТУРА

1. Армстронг М. Стратегическое управление человеческими ресурсами. Пер. с англ. – М.: ИНФРА-М. 2002.
2. Бейзер Б. Тестирование черного ящика. Технология функцио-нального тестирования программного обеспечения и систем. Пер. с англ. – СПб: ПИТЕР. 2004.
3. Блэк Р. Ключевые процессы тестирования. Пер. с англ. – М: ЛОРИ. 2006.
4. Боэм Б.У. Инженерное проектирование программного обеспечения. Пер. с англ./Под ред. А.А. Красилова. – М.: Радио и связь. 1985. (Barry W. Boehm. Software Engineering Economics. Prentice-Hall. 1981).
5. Вигерс К.И. Разработка требований к программному обеспечению. Пер. с англ. – М.: Русская редакция. 2004.
6. Волков О.И., Скляренко В.К. Экономика предприятия. – М.: ИНФРА-М. 2007.
7. Гецци К., Джазайери М., Мандриоли Д. Основы инженерии программного обеспечения. Пер. с англ. – СПб.: БХВ-Петербург. 2005.
8. Гринфилд Д., Шорт К. Фабрика разработки программ. Пер. с англ. – М: Диалектика. 2007.
9. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. Пер. с англ. – М. ЛОРИ. 2003.
10. Дмитриева М.А., Крылов А.А. Психология труда и инженерная психология. – М.: Дельфа. 2005.
11. Ильин А.И. Планирование на предприятии. Учебное пособие: В 2-х ч. Ч.1. Стратегическое планирование. – Минск: Новое знание, 2000.
12. Канер С., Фолк Д., Нгуен Е. Тестирование программного обеспечения. Пер. с англ. – М: ДиаСофт. 2001.
13. Кантор М. Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения. Пер. с англ. – М.: Вильямс. 2002.
14. Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. Пер. с англ. – М.: Вильямс. 2002.

15. Липаев В.В., Потапов А.И. Оценка затрат на разработку программных средств. – М.: Финансы и статистика. 1988.

16. Липаев В.В. Методы обеспечение качества крупномасштабных программных средств. – М.: РФФИ. СИНТЕГ. 2003.

17. Липаев В.В. Функциональная безопасность программных средств. – М.: СИНТЕГ. 2004.

18. Липаев В.В. Анализ и сокращение рисков проектов сложных программных средств. – М.: СИНТЕГ. 2004.

19. Липаев В.В. Программная инженерия. Методологические основы. Учебник. – М.: ТЕИС. 2006.

20. Липаев В.В. Тестирование крупных комплексов программ на соответствие требованиям. Учебник. – М.: Глобус. 2008.

21. Липаев В.В. Человеческие факторы в программной инженерии: рекомендации и требования к профессиональной квалификации специалистов. Учебник. – М.: СИНТЕГ. 2009.

22. Липаев В.В. Сертификация программных средств, Учебник. – М.: СИНТЕГ. 2010.

23. Липаев В.В. Тестирование компонентов и комплексов программ. Учебник. – М.: СИНТЕГ. 2010.

24. Липаев В.В. Экономика производства сложных программных продуктов. Издание второе. – М.: СИНТЕГ. 2011.

25. Мазер Г.Я., Мхитарян Н.А. Экономика машиностроения. – М.: Изд. МГОУ. 2008.

26. Организация производства и управления предприятием. Учебник. Под ред. О.Г. Туровца. – М.: ИНФРА-М. 2006.

27. Рекомендации по преподаванию программной инженерии и информатики в университетах. Пер. с англ. – М.: ИНТУИТ. 2007.

28. Скопин И.Н. Основы менеджмента программных проектов. – М.: ИНТУИТ.2004.

29. Соммервилл И. Инженерия программного обеспечения. 6-е издание. Пер. с англ. – М.: Вильямс. 2002.

30. Стрелков Ю. К. Инженерная и профессиональная психология. – М.: Академия. 2001.

31. Торингтон Д., Холл Л., Темлер С. Управление человеческими ресурсами. Учебник. Пер. с англ. – М.: Дело и сервис. 2004.

32. Трубачев А.П., Долинин М.Ю., Кобзарь М.Т. и др. Оценка безопасности информационных технологий. Общие критерии. Пер. с англ. Под ред. В.А. Галатенко. – М.: СИП РИА, 2001.

33. Тэллес М., Хсих Ю. Наука отладки. Пер. с англ. – М.: Кудиц-образ. 2003.

34. Уайт Б.А. Управление конфигурацией программных средств. Практическое руководство по Rational ClearCase. Пер. с англ. – М. ДМК Пресс. 2002.

35. Фатрелл Р.Т., Шафер Д.Ф., Шафер Л.И. Управление программными проектами: достижение оптимального качества при минимальных затратах. Пер. с англ. – М.: Вильямс. 2003.

36. Щербо В.К., Козлов В.А. Функциональные стандарты в открытых системах. Ч. 1. Концепция открытых систем. Ч.2. Международные функциональные стандарты. – М.: Изд. МЦНТИ. 1997.

37. Экономика промышленного предприятия. Учебник. Под ред. Е.Л. Кантора. – М.: МарТ. 2007.

38. Boehm B.W. et al. Software cost estimation with COCOMO II. Prentice Hall PTR. New Jersey. 2000.

39. Boehm B.W. Software risk management. IEEE Computer Society Press. Washington. 1989.

40. Charett R. Software engineering risk analysis and management. N.Y.: McGraw – Hill. 1989.

41. Davis A. Software requirements: Objects, functions and states. – Englewood Cliffs. NY. Prentice-Hall. 1993.

42. Jones C. Applied software measurement, assuring productivity and quality. McGraw-Hill. NY. 1996.

43. Higuera R., Haimes Y. Software risk management. Pittsburg. Software engineering institute, Cornegie Mellon University. – 1996.

44. Kit E. Software Testing in the Real World - Improving the Process. Addison-Wesley.1996.

45. Karolak D. W. Software engineering risk management. IEEE Computer Society Press. Washington. 1996.

46. Londeix B. Cost estimation for software development. Cornwall: Addison-Wesley. 1987.

47. Mutafelija B., Stromberg H. Systematic process improvement using ISO 9001:2000 and CMMI. SEI. 2003.

48. Quarterman J.S., Wilhelm S. Unix, Posix and open systems: The open standards puzzle. N.Y., Addison - Wesley. 1993.

49. Shooman M.L. Software Engineering: Reliability, Development and Management. N.Y. McGraw-Hill. 1983.